



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

POROVNÁVÁNÍ VÝZNAMNÝCH BODŮ PRO DETEKCI OBJEKTŮ V OBRAZE

IMAGE OBJECTS DETECTION BASED ON THE FEATURE POINTS MATCHING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VOJTĚCH TRÁVNÍČEK

VEDOUcí PRÁCE

SUPERVISOR

Ing. VRATISLAV HARABIŠ

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor

Biomedicínská technika a bioinformatika

Student: Vojtěch Trávníček

ID: 133988

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Porovnávání významných bodů pro detekci objektů v obraze

POKYNY PRO VYPRACOVÁNÍ:

1) Prostudujte a popište metody detekce významných bodů v obraze a zaměřte se rovněž na metody popisu těchto bodů. 2) Prostudujte metody porovnávání významných bodů a proveďte návrh metody porovnání významných bodů pro vyhledávání objektů. 3) Vytvořte vhodná testovací data pro ověření jednotlivých metod. 4) Metody ověřte a vytvořte program v programovém prostředí Matlab nebo v C++, který bude sloužit pro vyhledávání definovaných objektů v obraze na základě vyhledávání a porovnávání významných bodů. 5) Program dostatečně otestujte na modelových i reálných datech. 6) Proveďte zhodnocení dosažených výsledků.

DOPORUČENÁ LITERATURA:

[1] CHOI, O., KWEON, I.S. Robust feature point matching by preserving local geometric consistency. Computer Vision and Image Understanding. roč. 113, č. 6, s. 726-742, 2009.

[2] NAIN, N., LAXMI, V., BHADVIYA, B., DEEPAK, B.M., AHMED, M. Fast feature point detector. IEEE International Conference on Signal Image Technology and Internet Based Systems, s. 301-306, 2008.

Termín zadání: 11.2.2013

Termín odevzdání: 31.5.2013

Vedoucí práce: Ing. Vratislav Harabiš

Konzultanti bakalářské práce:

prof. Ing. Ivo Provazník, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce se týká oboru počítačového vidění. Konkrétně se věnuje extrakci významných bodů z obrazu jako prostředek pro srovnání obrazů a vyhledávání objektů. Jsou zde zmíněny 4 metody, které jsou porovnávány z hlediska účinnosti a využití. Jako hlavní algoritmy jsou zde zmíněny SIFT a SURF, které jsou v poslední době nejčastěji využívány. Jsou zde popsány taky metody popisu významných bodů a jejich porovnávání. Jsou zde vloženy testovací obrazy pro primární testování implementovaného algoritmu. Nakonec je popsána implementace metody SURF a je otestována z hlediska několika nejvýznamnějších parametrů.

Abstract

This paper is concerned in branch of computer vision. Methods for extracting feature points are presented as tools for image comparison and finding objects in images. Four methods are mentioned which are compared with respect to their effectiveness and utilization. Algorithms SIFT and SURF are described as a state-of-the-arts. This paper also mentions methods for describing feature points and their comparison. Testing images are inserted as a tool for first testing of implemented algorithm. Finally, the implemented method SURF is described and tested with respect to several most relevant parameters.

Klíčová slova

Počítačové vidění, detekce významných bodů v obraze, SIFT, SURF, porovnávání obrazů, testovací obrazy pro detektory významných bodů, porovnání medicínských obrazů, fundus kamera

Key words

Computer vision, feature point detection, SIFT, SURF, image comparison, feature point testing images, comparison of medical images, fundus camera

Bibliografická citace

TRÁVNÍČEK, V. *Porovnávání významných bodů pro detekci objektů v obraze*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 42s. Vedoucí bakalářské práce Ing. Vratislav Harabiš.

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma porovnávání významných bodů pro detekci objektů v obraze jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009Sb.

V Brně dne 30.5.2013

podpis autora:

Poděkování

Děkuji svému vedoucímu semestrální práce Ing. Vratislavu Harabišovi za cenné rady, připomínky a pomoc při řešení problémů vzniklých při psaní této práce.

Obsah

Seznam obrázků	2
Seznam grafů	4
Seznam tabulek	4
1 Úvod	5
2 Starší operátory	6
2.1 Moravcův operátor	6
2.2 Harrisův operátor	7
2.2.1 Využití	8
3 SIFT (Scale Invariant Feature Trans-form)	9
3.1 Popis algoritmu	9
3.1.1 Detekce extrémů ve scale-space	10
3.1.2 Zpřesnění lokalizace nalezených bodů	11
3.1.3 Rozhodnutí o orientaci	12
3.1.4 Tvorba deskriptorů	12
3.1.5 Hledání korespondencí	13
3.2 Využití	14
4 SURF (Speeded Up Robust Features)	15
4.1 Popis algoritmu	16
4.1.1 Extrakce významných bodů	16
4.1.2 Tvorba deskriptorů	17
4.2 Využití	18
5 Porovnávání významných bodů	18
6 Testovací obrazy	19
7 Implementace metody SURF	20
7.1 Programové prostředí	20
7.2 Popis programu	20
7.2.1 Konvoluce s Gaussovými filtry	21
7.2.2 Extrakce bodů	22

7.2.3	Zpřesnění lokalizace bodů 3D Taylorovým polynomem	24
7.2.4	Přiřazení orientace	25
7.2.5	Výpočet deskriptorů	25
7.2.6	Porovnání deskriptorů	26
8	Testování na modelových datech.....	26
9	Testování na reálných datech.....	28
9.1	Testování odolnosti vůči rotaci.....	30
9.2	Testování účinnosti při změně měřítka.....	32
9.3	Testování algoritmu při přidání multiplikativního šumu.....	35
9.4	Testování úspěšnosti detekce při deformacích obrazu	37
10	Závěr.....	39
	Seznam literatury.....	41
	Seznam použitých zkratek.....	43

Seznam obrázků

Obrázek 1: Výsledek detekce významných bodů Moravcovým detektorem na testovacím obrázku. (Převzato z [17])	6
Obrázek 2: Ukázka detekce významných bodů Harrisovým operátorem.....	8
Obrázek 3: Ukázka výpočtu DoG ze scale space (převzato z [14] a upraveno).....	10
Obrázek 4: Hledání extrémů ve scale space. Křížek značí zvažovaný bod, zeleně jsou značeny body, se kterými je vyšetřovaný bod porovnáván (převzato z [14] a upraveno).....	11
Obrázek 5: Deskriptor je tvořen gradienty vypočteny v okolí každého významného bodu, což je znázorněno vlevo. Ty jsou pak váženy Gaussovým oknem, které je zde symbolizováno modrým kruhem. Z každého regionu 4x4 je pak sestaven histogram orientací. Výsledný deskriptor je znázorněn vpravo, kde je zobrazen ve velikosti 2x2 histogramy, zatímco v algoritmu je použit deskriptor 4x4 vypočtený z 16x16 pixelů. Převzato z [14] a upraveno.	12
Obrázek 6: Významné body detekované algoritmem SIFT s jejich velikostí gradientů a orientacemi. Pro detekci byl použit software [13].....	14
Obrázek 7: Diskretizované derivace Gaussovy funkce a jejich aproximace. Vlevo nahoře je zobrazena druhá derivace Gaussovy funkce podle y, pod ní její aproximace. Vpravo nahoře je druhá derivace Gaussovy funkce ve směru xy a pod ní její aproximace. Šedé plochy jsou rovny nule. Převzato z [2].	17
Obrázek 8: Ukázky testovacích obrazů. Obrazy b) a c) jsou převzaty z [16].....	19
Obrázek 9: Schéma programu.....	21
Obrázek 10: Ukázka symetrického rozšíření obrázku jako prevence indexování mimo obraz	22
Obrázek 11: Grafická ukázka generování oktáv. Na ose x je zobrazeno měřítko, osa y reprezentuje oktávu. Čísla v jednotlivých kolonkách reprezentují velikost konvoluční masky. Převzato z [1] a upraveno.....	23
Obrázek 12: 2D Haarovy vlny, vlevo ve směru x, vpravo ve směru y. Bílá oblast představuje váhu -1, černá oblast váhu 1.	25
Obrázek 13: Ukázka natočení masek pro výpočet Haarových vln. V obou případech bylo použito natočení o 116°. Na obrázku vlevo je použita maska vztahující se k měřítku s=1, vpravo k měřítku s=10. Z důvodu limitace původním vzorkováním obrazu lze natočit konvoluční masku v malých měřítkách pouze o násobky 90°	26
Obrázek 14: Ukázka extrahování bodů v jednotlivých oktávách na základním obrázku se dvěma intenzitami (0 a 255). Na obrázku a) jsou zobrazeny body nalezeny v první oktávě, na obrázku b) jsou body extrahované z druhé oktávy a na obrázku c) pak ze třetí oktávy.....	27

Obrázek 15: Ukázka orientace jednotlivých bodů. Počátek šipek reprezentuje vždy významný bod. Natočení šipek odpovídá zjištěné orientaci významných bodů, poměr délek šipek pak odpovídá měřítku, ve kterém byly body nalezeny (poměr odpovídá vždy pouze na jednom obrázku, z důvodu lepšího zobrazení).	27
Obrázek 16: Ukázka extrahování bodů v jednotlivých oktávách. Na obrázku a) jsou body extrahované z první oktávy, na obrázku b) z druhé oktávy a na obrázku c) ze třetí oktávy.	28
Obrázek 17: Ukázka snímku z fundus kamery. Převzato z [18].	29
Obrázek 18: Ukázka snímku sítnice připraveného pro detekci.	30
Obrázek 19: Ukázka správné detekce na obrazu natočeném o 100° . Jako shoda bylo	31
Obrázek 20: Ukázka správné detekce při změně měřítka na 0,4.	33
Obrázek 21: Ukázka detekce při zašumění multiplikativním šumem s relativní směrodatnou odchylkou 0,07. Podle výše zmíněné tolerance je zde správně detekováno deset	35
Obrázek 22: Ukázka správné detekce při deformaci obrazu. Vnitřní úhel v kosodélníku je zde $73,3^\circ$, správně bylo spárováno 12 bodů, nesprávně žádný.	37

Seznam grafů

Graf 1: Závislost úspěšnosti detekce na úhlu natočení	32
Graf 2: Graf závislosti poměru úspěšných detekcí na použitém měřítku	34
Graf 3: Závislost úspěšnosti detekce na směrodatné odchylce šumu	36
Graf 4: Graf závislosti úspěšnosti detekce na ostrém vnitřním úhlu v kosodélníku deformovaného obrazu.	38

Seznam tabulek

Tabulka 1: Výsledky testování odolnosti detekce vůči rotaci obrázku	31
Tabulka 2: Výsledky testování odolnosti vůči rotaci	33
Tabulka 3: Výsledky testování úspěšnosti detekce při přidání multiplikativního šumu	36
Tabulka 4: Výsledné hodnoty vzešlé z testování úspěšnosti detekce při deformaci obrazu ..	38

1 Úvod

Počítačové vidění se dá charakterizovat jako schopnost počítače vidět a porozumět snímku. Říká se, že jeden obraz vypovídá více jak tisíc slov. Proto je hlavní myšlenkou naučit počítač analyzovat obrázek tak, aby nám o něm mohl říci co možná nejvíce. Využívá se v mnoha aplikacích jako např. automatické navádění robotů, fúze obrazů, rozpoznání obličeje, 3D rekonstrukce objektů atd. Obraz je v počítači uložen jako 2D rozhraní (matice) čísel, přičemž jedno číslo odpovídá jednomu pixelu. V této práci budeme pracovat pouze s šedotónovými obrazy, kdy je nominální hodnota pixelu v intervalu od 0 do 255, kde hodnota pixelu odpovídá stupni šedi (0 odpovídá černé, 255 bílé). Cíl počítačového vidění je zpracovat tato čísla tak, aby je počítač nechápal pouze jako sled čísel, ale jako obraz, na kterém se nachází určitý objekt, popř. rozpoznal činnost objektu či místo kde se nachází. Tato práce se zabývá konkrétně porovnáváním obrazů a vyhledáváním objektů v obraze. Hlavní myšlenkou těchto algoritmů je, že se nebudou zabývat celým obrazem, ale zaměří se pouze na takové oblasti, které jsou pro obraz typické. Ty se nazývají důležité body, jejichž nalezení, popis a porovnávání je hlavním předmětem této práce. Hlavním faktorem těchto bodů je jejich opakovatelnost (z angl. repeatability), což znamená, zda je detektor schopný najít ty samé body za různých zobrazovacích podmínek. Pro provedení rešerše jsem si vybral čtyři algoritmy, které jsou jedny z nejvíc používaných a jejichž články mají nejvíc citací, což svědčí o jejich kvalitě.

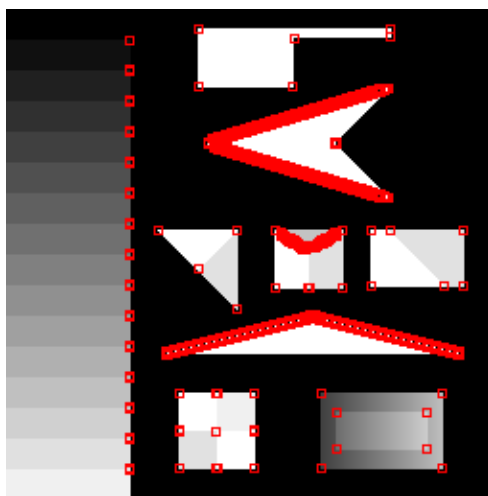
Harrisův detektor (Harris corner detektor) publikovaný v roce 1988 (viz [8]) vyhledává rohy v obraze. Rohy jsou body v obraze, kde se potkávají dvě hrany, přičemž jako hrany se označují místa s náhlou změnou intenzity v jednom směru. Dále jsem si zvolil detektor SIFT (Scale Invariant Feature Detector) navržený D. Lowem roku 2004. Součástí tohoto algoritmu je také deskriptor, který popisuje nalezené důležité body a metoda vzájemného porovnávání nalezených bodů. Prvky SIFT detektoru se inspiroval Herbert Bay, který v roce 2006 zveřejnil detektor SURF v publikaci [2]. Jeho výhodou je hlavně rychlost, která by měla být dostačující pro použití real-time.

2 Starší operátory

2.1 Moravcův operátor

Moravcův operátor je jeden ze starších detektorů, přesto má ještě dnes v různých modifikacích využití. Moravcův operátor používá okno malého rozměru (3×3 pixely), které se posunuje kolem vyšetřovaného bodu. Počítá průměrnou změnu intenzity, která vyplývá z posunování okna do různých směrů. V takovém případě bereme v potaz tři případy:

1. Pokud bude uvažovaná oblast obrazu plochá (tzn. zhruba o konstantní intenzitě), pak budou všechny posuny vykazovat pouze malou změnu.
2. Pokud okno narazí na hranu, potom všechny posuny podél hrany budou vykazovat malou změnu a naopak na posunech kolmých na hranu budeme pozorovat velkou změnu.
3. Pokud bude okno přecházet přes izolovaný bod, pak všechny posuny budou vykazovat velkou změnu. O roh se potom jedná, pokud minimální změna všech čtyř posuvů je velká (nad určitým prahem).



Obrázek 1: Výsledek detekce významných bodů Moravcovým detektorem na testovacím obrázku. (Převzato z [17])

Na obrázku 1 je výsledek detekce významných bodů, kdy byl práh určen experimentálně, aby detekoval co nejspolehlivěji všechny rohy na obrázku. Moravcův detektor však trpí několika neduhy. Na obrázku je poznat, že odezva detektoru je příliš silná na diagonálních hranách. Dalším z nedostatků je velká citlivost na šum z důvodu použití čtvercového okna, které není nijak váhováno, tudíž při výpočtu mají všechny hodnoty stejnou

váhu nezávisle na vzdálenosti od vyšetřovaného bodu. Naopak, vzhledem ke své výpočetní nenáročnosti je hlavním kladem rychlost.

Důkaz, že Moravcův operátor má své využití i v jednadvacátém století poskytuje publikace [10], ve které M. Sharmila Kumari a B. H. Shekar zveřejnili detektor textu z obrazu či videa za použití právě Moravcova operátoru. Využili faktu, že v textovém poli je hodnota Moravcova operátoru daleko větší než v pozadí, což je dáno větším počtem velmi výrazných hran, které způsobují jednotlivá písmena. Podobným způsobem detekuje text také lidský mozek. Autoři popisují detektor textu jako velmi jednoduchý na implementaci s výsledky srovnatelnými s ostatními.

2.2 Harrisův operátor

Harrisův operátor byl navržen roku 1988 Chrisem Harrisem a Mikem Stephensem. Vychází z Moravcova detektoru, kdy autoři v publikaci [8] rozebírají jeho nedostatky, na které následně navrhli řešení. Operátor tvoří symetrická matice, která úzce souvisí s autokorelací. Z matice se pak počítají vlastní čísla α a β , přičemž platí 3 kritéria:

1. Obě vlastní čísla jsou malá, tedy autokorelační funkce je plochá, pak část obrazu pod zvažovaným oknem má přibližně konstantní intenzitu.
2. Jedno z vlastních čísel je malé zatímco druhé velké, což značí hranu, protože posun podél hrany indikuje malé vlastní číslo, zatímco posun kolmo na hranu značí vlastní číslo o vysoké hodnotě.
3. Obě vlastní čísla jsou velká, to je způsobeno projitím autokorelační funkce přes pík, což značí roh.

Pro získání významných bodů je tedy potřeba zhodnotit parametry α a β . Abychom se ale vyhnuli počítání vlastních čísel matice, je výhodné použít $\text{Tr}(M)$ (stopa matice z angl. trace) a $\text{Det}(M)$ (determinant matice) [8]:

$$\text{Tr}(M) = \alpha + \beta = A + B, \quad (2.1)$$

$$\text{Det}(M) = \alpha\beta = AB - C^2, \quad (2.2)$$

,kdy M je symetrická matice ve tvaru

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}. \quad (2.3)$$

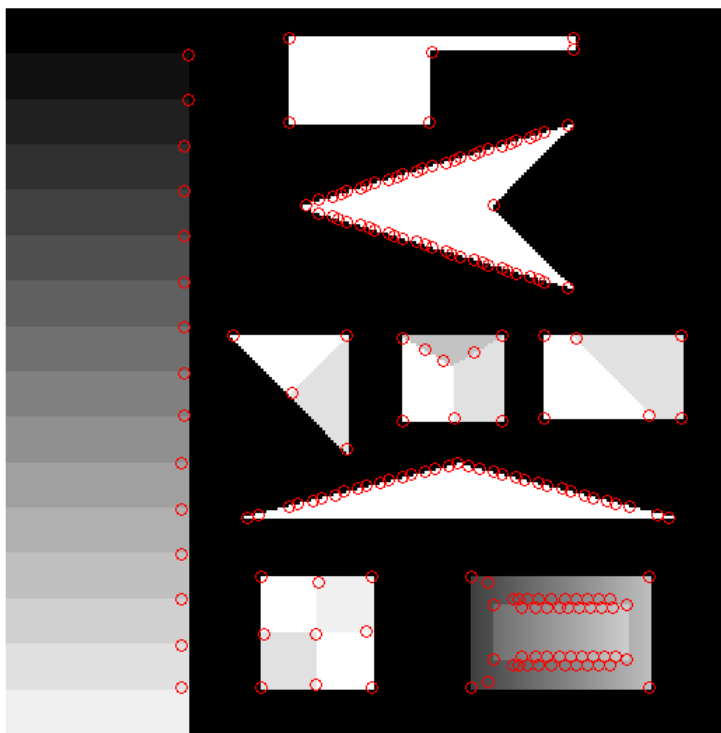
Z determinantu a stopy matice potom vypočítáme proměnnou R , podle které určíme, zda se jedná o hranu roh, či plochu [8]:

$$R = \text{Det} - k * \text{Tr}^2. \quad (2.4)$$

Pokud je R kladné, jedná se o roh, záporné hodnoty indikují hranu a malé hodnoty poukazují na plochu o konstantní intenzitě. O tom co je dostatečně velká hodnota pro to, aby šlo prohlásit odezvu za důležitý bod rozhodneme prahováním. Pokud bude při aplikaci

potřeba hodně důležitých bodů, zvolíme práh nižší, pokud si vystačíme s menším počtem detekovaných bodů avšak s větší stabilitou, zvolíme práh vyšší.

Harrisův detektor je na svou dobu velmi stabilní, je odolný vůči šumu i vůči rotaci. Jeho nedostatek spočívá v závislosti na měřítku, což je pro aplikací v reálném světě velký nedostatek. Přesto ale dodnes nalézá využití v mnoha aplikacích, kde nezávislost na měřítku není podmínkou.



Obrázek 2: Ukázka detekce významných bodů Harrisovým operátorem

2.2.1 Využití

Názorným příkladem využití Harrisova operátoru je v publikaci [21], kde autoři použili významné body extrahované právě Harrisovým detektorem pro odhad pohybu tumoru při ozařování. Při léčbě tumoru ozařováním totiž nastává problém s přesným zaměřením nádorového ložiska, které nikdy není ve stejné pozici oproti externím značkám na kůži pacienta. Tato pozice se může měnit z různých důvodů. Opomeneme-li pohyb pacienta, který se dá značně omezit, zůstávají nám ještě faktory, jako jsou respirační pohyby či pohyby vznikající plněním tkání krví. Proto by ideální radioterapie měla tyto nepřesnosti zvážit a přizpůsobit radiační paprsek pozici tumoru. Návrh na řešení tohoto problému přinesli roku 2009 ZOU Xiao-chun a HE Ming-yi v jejich práci [21], kde se zabývají sledováním tumoru na základě obrazových dat (jako reprezentativní vzor byla použita CT sekvence jater o 256 stupních šedi). Algoritmus se skládá ze tří částí, přičemž první je právě extrakce významných bodů Harrisovým operátorem, dále pak shromáždí body do skupin minimalizací SAD (z angl. sum of absolute differences) a v poslední sekci pak odhaduje pozici v dalším snímku na

základě posledních dvou snímků. Tyto postupy jsou však nad rámec této práce. Experimentální výsledky vykazují dostatečnou kvalitu na to, aby byla metoda zavedena do medicínské praxe.

3 SIFT (Scale Invariant Feature Transform)

Sift detektor byl navržen Davidem Lowem roku 2004 v University of British Columbia, kde byl také patentován. Publikace, ve které byl zveřejněn [14] má přes 1500 citací a v poslední době patří mezi nejdůležitější práce v oboru počítačového vidění. Hlavní výhodou tohoto detektoru je nezávislost na měřítku, proto se dá použít na porovnání obrazů, které byly pořízeny v různém úhlu (do určité meze) a z různé vzdálenosti od scény. Detektor vykazuje také velkou odolnost vůči šumu a změně osvětlení. Zajímavostí je, že postup detekce důležitých bodů a následného porovnávání mezi obrazy je velmi podobný fyziologickým procesům lidského mozku.

3.1 Popis algoritmu

Jak už jsem uvedl, hlavní výhodou algoritmu SIFT je nezávislost na měřítku. Toho je dosaženo hledáním významných bodů v tzv. scale-space (podrobněji v [11]). Jak víme zachycením 3D scény na 2D obraz ztrácíme přehled o třetí dimenzi, kterou můžeme charakterizovat jako měřítko. Abychom při detekci významných bodů dosáhli nezávislosti na měřítku, musíme pak zohlednit všechna možná měřítka, tedy nalézt ve 2D obrazu údaj o třetí dimenzi. Nositel tohoto údaje je frekvenční spektrum daného obrazu. Nejvyšší frekvence nesou údaje o nejjemnějších detailech, tedy o nejmenším měřítku, naopak vyšší frekvence jsou nosiče základních rysů obrazu tedy největším měřítku. Pro detekci napříč více měřítky je tedy potřeba sestavit třetí dimenzi, tedy údaj o prostorové frekvenci. Toho lze dosáhnout opakovaným vyhlazováním obrazu pomocí Gaussovy funkce. Ta se totiž chová jako dolní propust ve frekvenční oblasti, tudíž postupně odstraňuje vyšší frekvence, čímž dostává obrazy o menším měřítku. Gaussova funkce má tvar [14]

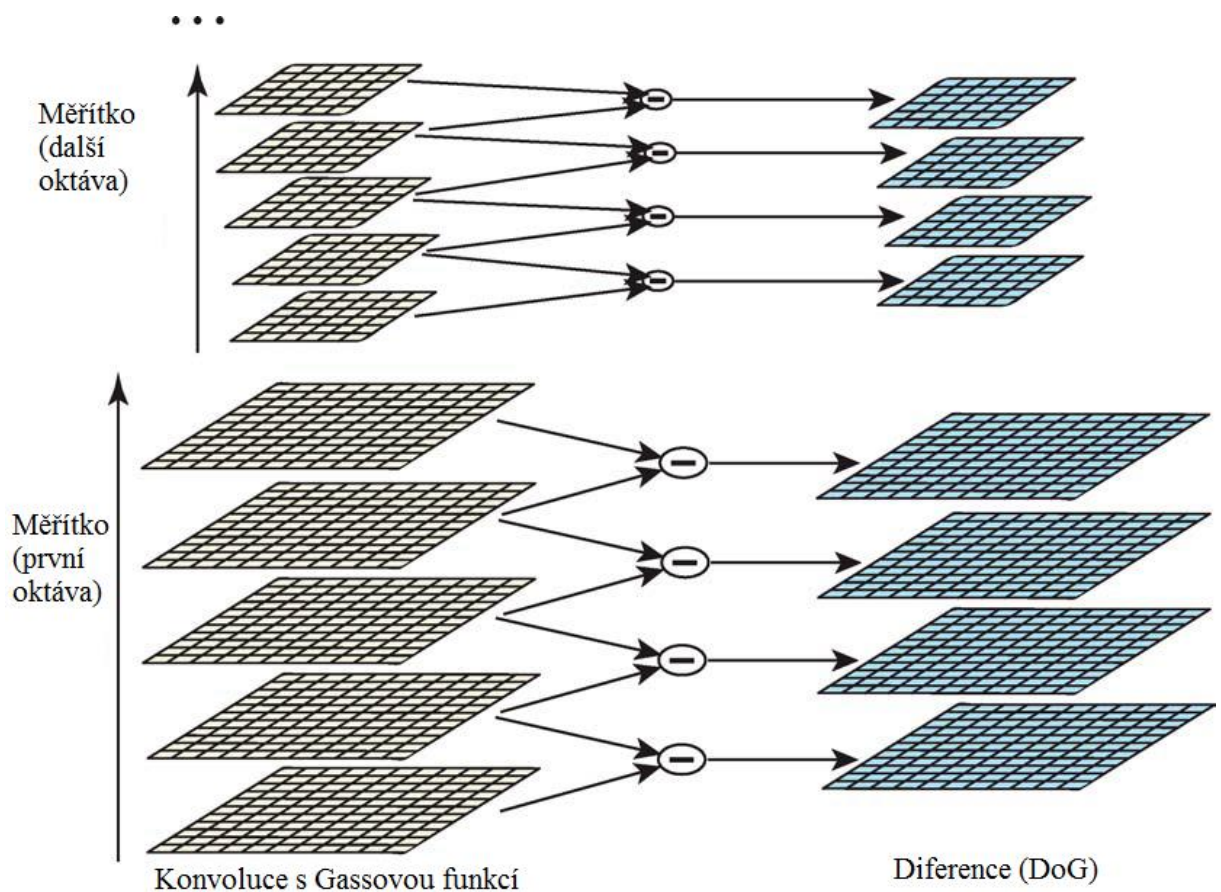
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \quad (3.1)$$

Kde x a y jsou souřadnice pixelu v obraze a parametr σ určuje šíři Gaussovy funkce, což lze při naší analogii vysvětlit jako mezní frekvence horní propusti nebo stupeň vyhlazení. Opakovaným filtrováním původního obrazu pak dostaneme více obrazů o různých měřítkách. Sestavením těchto obrazů postupně od největšího po nejmenší měřítko pak získáme 3D prostor $L(x, y, \sigma)$, tedy jednu oktávu prostoru zvaného scale-space (viz obrázek 2). Pro výpočet

dalších oktáv jsou v tomto algoritmu vybrány dva poslední obrazy z předešlé oktávy s největším vyhlazením neboli s největší hodnotou σ . Ty jsou pak podvzorkovány na polovinu a jsou základem pro další oktávu. Významné body jsou pak vyhledávány v tomto scale-space. Tím je dosažena nezávislost detektoru na měřítku, to vše ale na úkor výpočetní náročnosti. Autor David Lowe rozděluje svůj algoritmus na 4 části, a sice detekci extrémů ve scale space, zpřesnění lokalizace významných bodů, rozhodnutí o orientaci bodů a tvorbu deskriptorů. Této osnovy se budu držet i v této práci.

3.1.1 Detekce extrémů ve scale-space

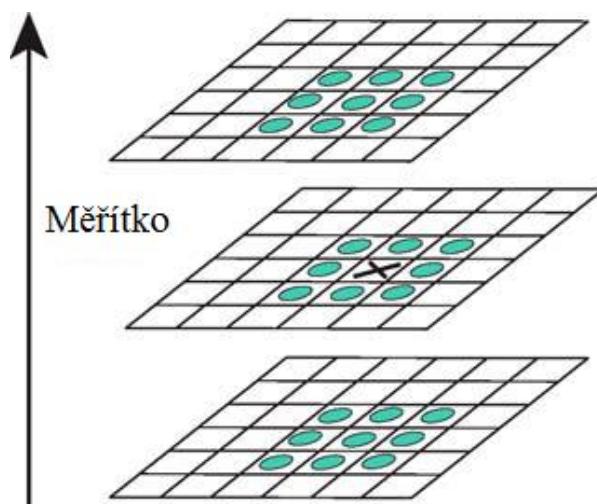
Jako základ pro detekci významných bodů použili autoři rozdíl obrazů vyhlazených Gaussovou funkcí DoG (z angl. Difference of Gaussian). Ten je počítán v každé oktávě ze sousedních obrazů.



Obrázek 3: Ukázka výpočtu DoG ze scale space (převzato z [14] a upraveno)

Existuje několik důvodů, proč si D. Lowe vybral právě funkci DoG. Diference je v tomto případě velmi výhodná funkce, jelikož může být spočítána z vygenerovaného scale space pouhým odečtením sousedních obrazů, což je velmi jednoduchá operace na výpočet. Dalším hlavním důvodem je, že DoG je blízkou aproximací LoG (Laplacian of Gaussian), což je operátor, který podle dřívějšího výzkumu poskytuje robustnější významné body než předešlé operátory, jako jsou použití gradientu, Hessovy matice či Harrisového operátoru.

Z výsledné modifikace scale space pomocí DoG (obrázek 2 vpravo) se potom detekují extrémy v každé oktávě. Za extrém je považován bod, který je maximem či minimem v jeho okolí, přičemž za okolí se zde považuje sousedních 8 bodů v daném obrazu a 9 bodů z měřítka nad a pod zvažovaným obrazem. Vyšetřovaný bod je tedy porovnáván se sousedními 26 pixely (viz obrázek 3).



Obrázek 4: Hledání extrémů ve scale space. Křížek značí zvažovaný bod, zeleně jsou značeny body, se kterými je vyšetřovaný bod porovnáván (převzato z [14] a upraveno).

3.1.2 Zpřesnění lokalizace nalezených bodů

Po nalezení lokálních extrémů metodou porovnávání se sousedními pixely je dalším krokem zhodnocení každého bodu, tedy zpřesnění jeho lokalizace a porovnání stability. Body s nízkým kontrastem (tedy citlivé na šum a tím nestabilní) pak mohou být odstraněny. Mathew Brown vynalezl metodu pro zpřesnění lokalizace významných bodů na subpixelovou úroveň. Na bod a jeho okolí aplikuje 3D kvadratickou funkci formou, kdy zpřesněná poloha klíčového bodu se určí jako extrém funkce. Prahováním extrému se navíc můžeme zbavit nízkokontrastních bodů, které jsou citlivé na šum a tím by snižovaly spolehlivost celého detektoru. Experimenty bylo dokázáno, že tento postup přinesl značné zlepšení stability jednotlivých bodů i procesu porovnávání obrazů.

Nevýhodou funkce DoG je, že můžeme nalézt extrémy podél hran, které jsou ale nepřesně lokalizovány a tím citlivé na šum. Proto je dalším krokem pro zvýšení stability detektoru odstranění bodů, které byly lokalizovány podél hran. To provádíme podobným způsobem jako u Harrisova detektoru, tedy použitím Hessianovy matice a jejich vlastních čísel. Použitím těchto postupů pak značně eliminujeme množství detekovaných bodů, přičemž nám zůstanou pouze stabilní body určené se subpixelovou přesností, čímž nám usnadní další kroky v porovnávání obrazů.

3.1.3 Rozhodnutí o orientaci

Přiřazení orientací významným bodům je nezbytným krokem pro dosažení nezávislosti na rotaci. Pro každý bod získaný postupem popsaným výše se vybere již vygenerované měřítko, kterému je bod nejbližší, aby se zachovala nezávislost na měřítku. V tomto obrázku $L(x,y)$ se pak spočítá velikost gradientu $m(x,y)$ a orientace $\theta(x,y)$ pro každý pixel:

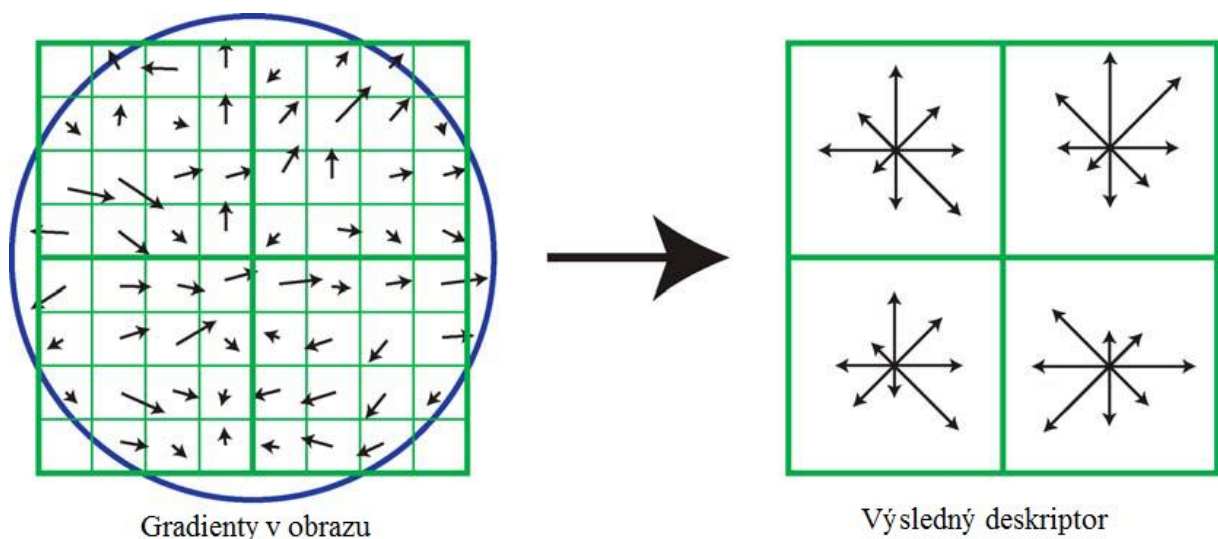
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \quad (3.2)$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))). \quad (3.3)$$

S gradientů vypočtených v okolí významného bodu se vytvoří histogram o 36 binech, kdy každý bin odpovídá 10° úhlu. Každý gradient přidaný do histogramu je vážen Gaussovským oknem s σ 1,5krát větší než σ daného měřítka. V histogramu je pak detekován největší pik, který je vybrán jako orientace významného bodu. Pokud je dále detekován pik o velikosti větší než 80% maxima, je vytvořen další významný bod na stejné pozici s orientací daného píku.

3.1.4 Tvorba deskriptorů

Předchozí operace braly v potaz měřítko, pozici a přiřadily orientaci každému významnému bodu. Tím že byly tyto vlastnosti analyzovány a popsány se dosáhlo nezávislosti na těchto parametrech. Dalším krokem je vytvoření deskriptoru, který by dosáhl nezávislosti na dalších parametrech, jako jsou změna osvětlení či místa při pořizování snímku 3D scény. Dřívější výzkum prokázal, že pro rozpoznání 3D objektu je nejlepší popis bodů pomocí gradientů. Na podobném principu totiž funguje i biologický proces vidění. Výpočet deskriptorů je znázorněn na obrázku 5.



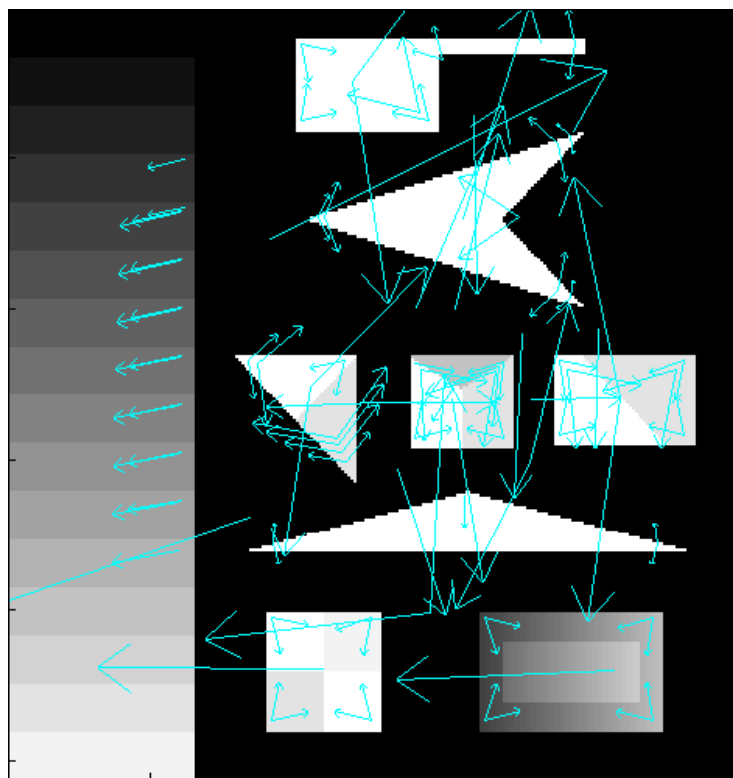
Obrázek 5: Deskriptor je tvořen gradienty vypočteny v okolí každého významného bodu, což je znázorněno vlevo. Ty jsou pak váženy Gaussovým oknem, které je zde symbolizováno modrým kruhem. Z každého regionu 4x4 je pak sestaven histogram orientací.

Výsledný deskriptor je znázorněn vpravo, kde je zobrazen ve velikosti 2x2 histogramy, zatímco v algoritmu je použit deskriptor 4x4 vypočtený z 16x16 pixelů. Převzato z [14] a upraveno.

Jak můžeme vidět na obrázku, v okolí bodu jsou vypočteny velikosti gradientů a orientace podle vzorců (3.2) a (3.3). Aby bylo dosaženo nezávislosti na rotaci, jsou tyto parametry počítány vzhledem k orientaci významných bodů. Z 4x4 podoblastí jsou vytvořeny histogramy o 8 binech (1 bin odpovídá 45°). Gradienty se v jednotlivých binech sumují, přičemž jsou váženy Gaussovskou funkcí, aby se předešlo náhlým změnám deskriptoru ovlivněnými gradienty vzdálenými od významného bodu. Hodnoty ve výsledném deskriptoru jsou pak normalizovány, čímž se dosáhne nezávislosti na změně kontrastu. Výsledný deskriptor bude tedy tvořen 16ti histogramy o 8 binech, což je celkem deskriptor o velikosti 128 binů.

3.1.5 Hledání korespondencí

Zatím jsem se zabýval pouze hledáním významných bodů v obraze jako základní předmět pro rozpoznávání obrazu. Posledním krokem celého procesu je hledání korespondencí mezi nalezenými body v různých obrazech. V tomto algoritmu je každý bod reprezentován svým deskriptorem o 128 hodnotách, proto je tedy hlavní úlohou hledat podobnost mezi těmito vektory. Hledání nejpodobnějšího deskriptoru probíhá počítáním Euklidovských vzdáleností, tedy kandidát z nejnižší vzdáleností je zvolen jako nejvhodnější. Jenže mnoho bodů detekovaných v obraze nebude mít shodu s předloženým vzorem, proto se hledal způsob jak tyto body eliminovat. Autoři [14] navrhli porovnání 2 kandidátů s nejmenší Euklidovskou vzdáleností tím, že je dali do poměru nejbližší ku druhému nejbližšímu. Výzkum ukázal, že pokud je poměr vyšší jak 0,8 je velká pravděpodobnost, že dojde k nesprávnému spárování. Proto je zde zaveden tento práh, což vede k odstranění 90% nesprávných shod a omezení pouze 5% správných shod. Porovnávání každého prvku proti celé databázi je ale výpočetně velmi náročné, proto je zde použito algoritmu BBF (Best-Bin-First), který je však nad rámec této práce. Pro podrobnější informace o tomto algoritmu odkazuji na zdroj [3]. Dále je zde použito Houghovy transformace pro hledání pozice objektu. Zde opět odkážu na zdroj [7].



Obrázek 6: Významné body detekované algoritmem SIFT s jejich velikostí gradientů a orientacemi. Pro detekci byl použit software [13].

3.2 Využití

Metoda SIFT se používá v mnoha aplikacích a má několik upravených verzí. Jedna z nich se nazývá n-SIFT a je to modifikace pro detekci významných bodů v n dimenzionálním prostoru. Tato metoda byla testována na medicínských 3D datech získaných z magnetické resonance a také 4D datech získaných z CT. Toto testování prováděl Warren Cheung na University of British Columbia a výsledky popsal v publikaci [9]. Nevýhodou počítání ve vícedimenzionálním prostoru se stala výpočetní náročnost, kdy v [9] uvádí výpočet dlouhý necelých 19 minut na datech, které obsahovaly 874 tisíc voxelů. Při aplikaci tohoto algoritmu ve více dimenzionálním prostoru musíme také počítat s rostoucí velikostí deskriptorů, což má také vliv na rychlost výpočtu.

Další využití v medicíně může mít SIFT při fúzi obrazů. V publikaci [12] použili autoři významné body extrahované právě metodou SIFT pro fúzi obrazů z ultrazvuku. Velkým nedostatkem ultrazvuku je nemožnost zobrazení většího pole, což je limitováno rozměry sondy. Tento problém je řešen zachycením většího počtu obrazů a jejich následné fúze. Jelikož SIFT už byl úspěšně použit pro tvoření panoramat, autoři [12] použili tento algoritmus také u ultrazvuku. Metoda byla testována na několika sadách obrazů, kde bylo dosaženo srovnatelných výsledků s dosavadními metodami. Nevýhodou této metody byla opět výpočetní náročnost, která byla větší ve srovnání s jinými metodami. Naopak hlavní výhodou byla nezávislost na rotaci. Při pořizování sady obrazů je sonda posouvána podél

vyšetřované tkáně. Pokud je sonda při tomto pohybu natáčena, vznikají obrazy s různou orientací. Při fúzi takovýchto obrazů pak vykazoval SIFT lepší výsledky než ostatní metody.

4 SURF (Speeded Up Robust Features)

SURF je metoda na extrakci významných bodů, jejich popis a porovnání. Metoda byla popsána roku 2006 Herbertem Bayem a spol. v [2]. Jak už název napovídá, hlavní výhodou tohoto algoritmu je rychlost. Té bylo dosaženo především použitím integrálního obrazu při konvolucích. Autor také využil znalostí předchozích detektorů (použití Hessovy matice, formování deskriptorů na základě histogramů), které se se snažil co nejvíce zjednodušit. K rychlosti také přispěly aproximace derivací Gaussových funkcí, které mohou být pomocí integrálního obrazu spočítány daleko rychleji.

4.1 Popis algoritmu

Při navrhování metody SURF [2] si autoři kladli za důraz navrhnout detektor a deskriptor rak, aby výrazně snížili výpočetní náročnost při zachování výkonu. Pro zvýšení výpočetní rychlosti použili autoři integrálního obrazu. Bod v integrálním obrazu $I_{\Sigma}(\mathbf{x})$ na pozici $\mathbf{x} = (x, y)$ je definován jako suma všech pixelů vstupního obrazu $I(i, j)$ tvořící obdélník mezi bodem \mathbf{x} a počátkem. Vyjádřeno rovnicí

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \quad (4.1)$$

V integrálním obrazu $I_{\Sigma}(\mathbf{x})$ je pak možné v jakémkoli čtvercovém regionu o libovolné velikosti spočítat sumu všech pixelů pouze sečtením čtyř čísel [20], což výrazně snižuje výpočetní náročnost.

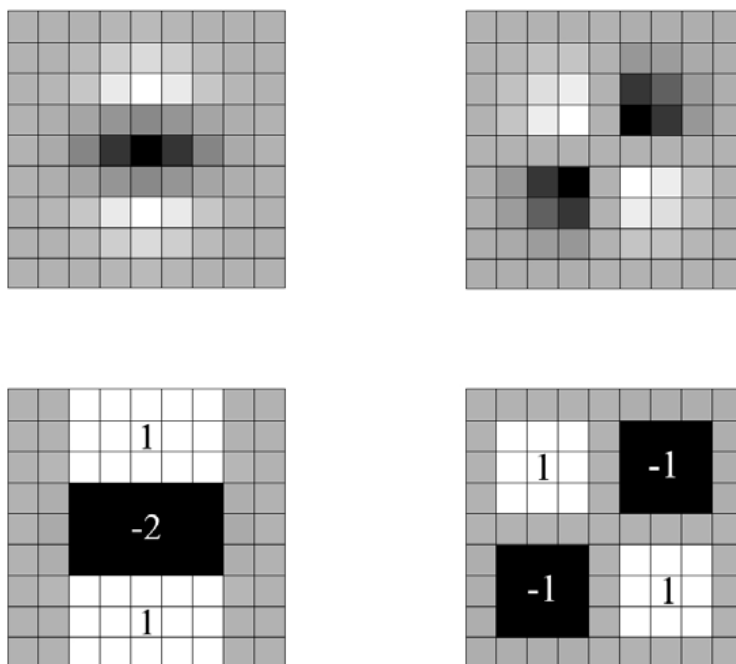
4.1.1 Extrakce významných bodů

Pro vyhledávání významných bodů si autoři [2] zvolili Hessovu matici. Ta byla použita již v několika detektorech včetně Harrisova (viz kapitola 2.2.). Hessova matice má tvar

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{pmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{pmatrix}, \quad (4.2)$$

kde \mathbf{x} značí souřadnice bodu obrazu o měřítku σ . $L_{xx}(\mathbf{x}, \sigma)$ je pak konvoluce druhé derivace Gaussovy funkce s originálním obrazem I v bodě \mathbf{x} , stejně tak $L_{xy}(\mathbf{x}, \sigma)$ a $L_{yy}(\mathbf{x}, \sigma)$. Nastává zde tedy otázka výpočtu druhých derivací Gaussovy funkce.

Gaussova funkce je optimální funkcí pro generování scale space. Z důvodu zjednodušení výpočtů zde není scale space generováno postupnými konvolucemi Gaussianu s originálním obrazem, ale jsou zde použity aproximace přímo jeho druhých derivací použitých v rovnici (4.2), což opět přispívá ke zrychlení výpočtu. H. Bay se zde odvolává na minulé úspěchy D. Lowe při aproximaci Laplaciánu rozdílem Gaussových funkcí.



Obrázek 7: Diskretizované derivace Gaussovy funkce a jejich aproximace. Vlevo nahoře je zobrazena druhá derivace Gaussovy funkce podle y , pod ní její aproximace. Vpravo nahoře je druhá derivace Gaussovy funkce ve směru xy a pod ní její aproximace. Šedé plochy jsou rovny nule. Převzato z [2].

Tyto aproximace uvedené v obrázku 6 jsou rozměru 9×9 a odpovídají nejmenšímu měřítku (tzn. největšímu prostorovému rozlišení). Další měřítka jsou pak generovány postupným zvětšováním konvoluční masky, která se aplikuje stále na originální integrální obraz. Tímto způsobem spočítány všechny prvky Hessovy ve všech měřítkách. Harris ve svém detektoru využíval determinant a stopu matice, zatímco H. Bay se spoléhá pouze na determinant, čímž opět snížil výpočetní náročnost. Po vypočtení determinantu jsou pak významné body hledány stejným způsobem jako u algoritmu SIFT, tedy bod je označen jako významný, pokud je maximem ve svém $3 \times 3 \times 3$ okolí.

4.1.2 Tvorba deskriptorů

Pro nezávislost deskriptoru na rotaci jsou přiřazeny orientace pro každému významnému bodu. Ty jsou opět počítány pomocí konvoluce, aby se využilo integrálního obrazu a zvýšila se rychlost výpočtu. Jako konvoluční masky se používají Haarovy vlny ve směru x a y . Pro rozhodnutí o orientaci se zvažuje kruhová oblast kolem každého bodu, ve které jsou zaznamenávány odezvy na Haarovy vlny do histogramu o 6ti binech (1bin odpovídá 60°), přičemž odezvy jsou váhovány Gaussovou funkcí umístěnou ve vyšetřovaném bodě. V histogramu pak největší pík reprezentuje orientaci významného bodu. Ostatní jsou zanedbány.

Pro formování deskriptoru se pak zvolí čtvercová oblast kolem významného bodu. Čtverec je natočen podle orientace významného bodu a jeho velikost je přizpůsobena měřítku, ve kterém se významný bod nachází. Ten je pak rozdělen na 4×4 podoblasti. V každé podoblasti je vybráno 5 pravidelně rozmístěných bodů, na které se aplikuje filtr stejný jako při přiřazování orientací (Haarovy vlny). Odezvy na tyto filtry ale už nemohou být počítány pomocí integrálního obrazu z důvodu natočení čtvercové oblasti zájmu. Necht' d_x a d_y jsou odezvy na Haarovy vlny ve směru x a y (směr je určen vzhledem k natočení čtvercové oblasti). Pro každou z 16 podoblastí jsou pak spočítány 4 parametry ($\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|$) tedy sumy odezev na Haarovy vlny. Výsledný deskriptor je tvořen $4 \times 4 \times 4$ elementy tedy má velikost 64binů.

4.2 Využití

V roce 2012 byla zveřejněna publikace [20], ve které autoři srovnávali metody SIFT a SURF z hlediska klasifikace medicínských obrazů. Úkolem detektorů bylo rozpoznat plíce na snímcích rentgenu. Bylo použito 65 trénovacích obrazů, přičemž algoritmy byly odzkoušeny na 4241 rentgenových snímků. Jako kritérium pro určení zda jsou na snímku plíce či ne se zvolil práh 0,2, a sice že minimálně 20% extrahovaných bodů musí spadat do kladné trénovací kategorie. Výsledky ukázaly, že mezi metodami SURF a SIFT je výrazný nepoměr. Přesto, že se použilo vcelku málo trénovacích obrazů, algoritmus SURF klasifikoval správně více než 96% snímků, zatímco SIFT necelých 57%. Autoři zhodnotili detektor SURF jako potenciálně silný nástroj k využití pro klasifikaci medicínských dat.

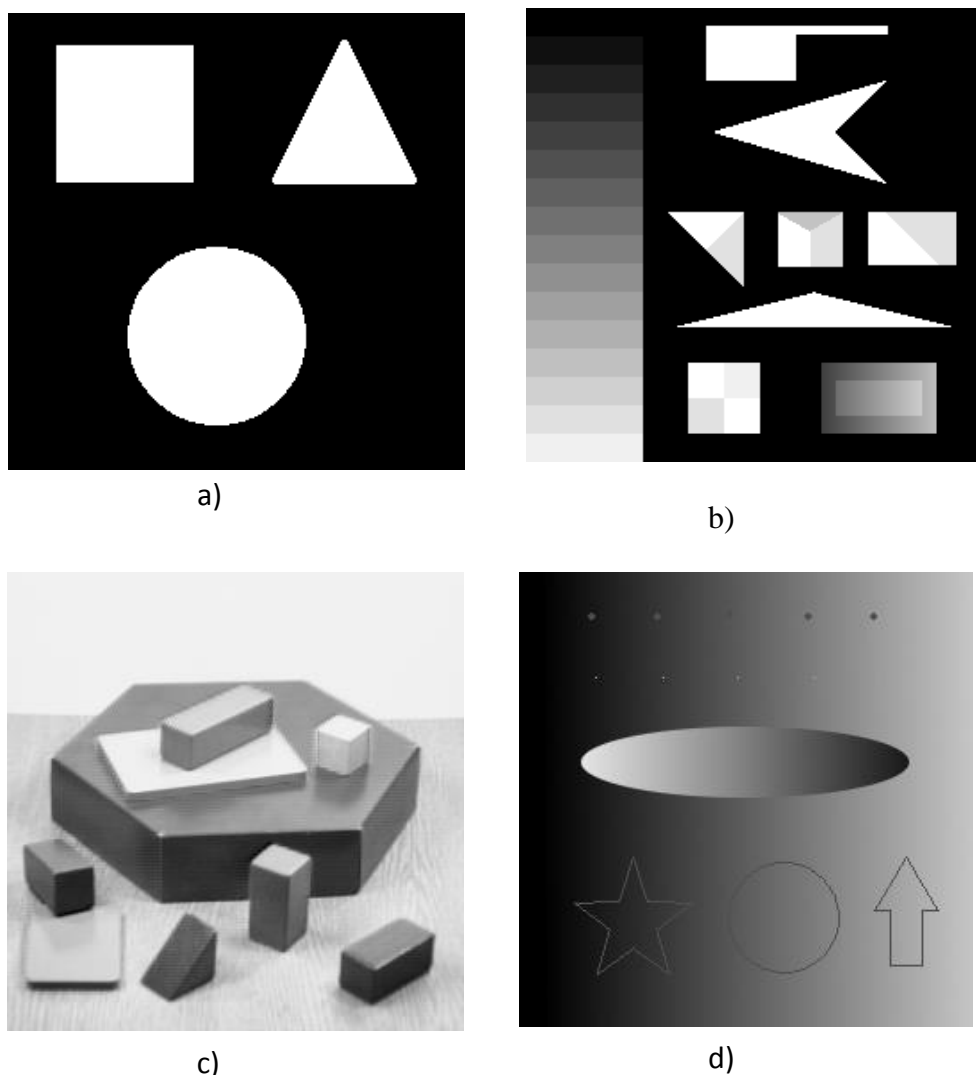
5 Porovnávání významných bodů

Doposud jsem se zabýval pouze extrakcí významných bodů a jejich popisem. Výjimkou je pouze kapitola 3.1.5, kde jsem popsal postup hledání korespondencí mezi nalezenými významnými body, který je součástí metody SIFT. Tento algoritmus je založený na počítání vzdáleností mezi vektory. Vzdálenost mezi vektory (deskriptory) je použita ve více algoritmech, ať už Euklidovská či Mahalanobisova. Zde však nastává problém, pokud má významný bod dva nejbližší kandidáty nebo pokud je nejbližší vzdálenost k bodu z druhého obrazu příliš velká. Tento problém je buď řešen aplikováním metody prahováním, nebo D. Lowe [14] vzal v potaz další bod s druhou nejmenší vzdáleností, kdy porovnával poměr mezi těmito dvěma kandidáty. Další možností je aplikování korelace. Toho využil v publikaci [22] Ji Zhou. Jako významné body si zvolil rohy detekované Harrisovým operátorem. Nepoužil žádného deskriptoru, nýbrž porovnával jejich okolí pomocí korelace. Ta vybere kandidáty na shodu, které jsou pak eliminovány dalšími matematickými postupy. Pak jsou tu možnosti využití neuronových sítí. Pro ty je ale potřeba větší množství učebních prvků, aby se sítě správně nastavily. Na principu neuronových sítí pak pracuje algoritmus SVM (Support Vector

Machine) [19], který byl použit při srovnávání algoritmů SIFT a SURF [20] a zmínil jsem ho v kapitole 4.2.

6 Testovací obrazy

Testovací obrazy se používají pro první zhodnocení detekce významných bodů. Při detekci na reálných obrazech lze jen těžko posoudit, zda detektor správně odpovídá na základní rysy obrazu, jako jsou hrany, rohy či gradienty. Proto jsou brány obrazy, kde jsou tyto prvky na první pohled rozeznatelné nebo jsou uměle vytvořeny tak, aby tyto základní rysy obsahovaly.



Obrázek 8: Ukázky testovacích obrazů. Obrazy b) a c) jsou převzaty z [16]

Obrázek 8 a) slouží pro testování nejzákladnějších vlastností detektoru. Obsahuje pouze intenzity 0 a 255 a úhly 90° a 60° . Obrázek 8 b) už obsahuje více úhlů a hlavně více intenzit. Zde se dá potom hodnotit co je pro detektor významnější, zda hrana s velkým

přechodem intenzit či roh s nižším přechodem intenzit. Obrázek 8 c) je často používán ve smyslu testování detektorů, je potom možné porovnat výsledky aplikace různých algoritmů na stejném obrázku, což je přínosné pro srovnání. Obrázek 8 d) pak obsahuje různé gradienty, které mají taky velký vliv na vyhledávání a popis významných bodů.

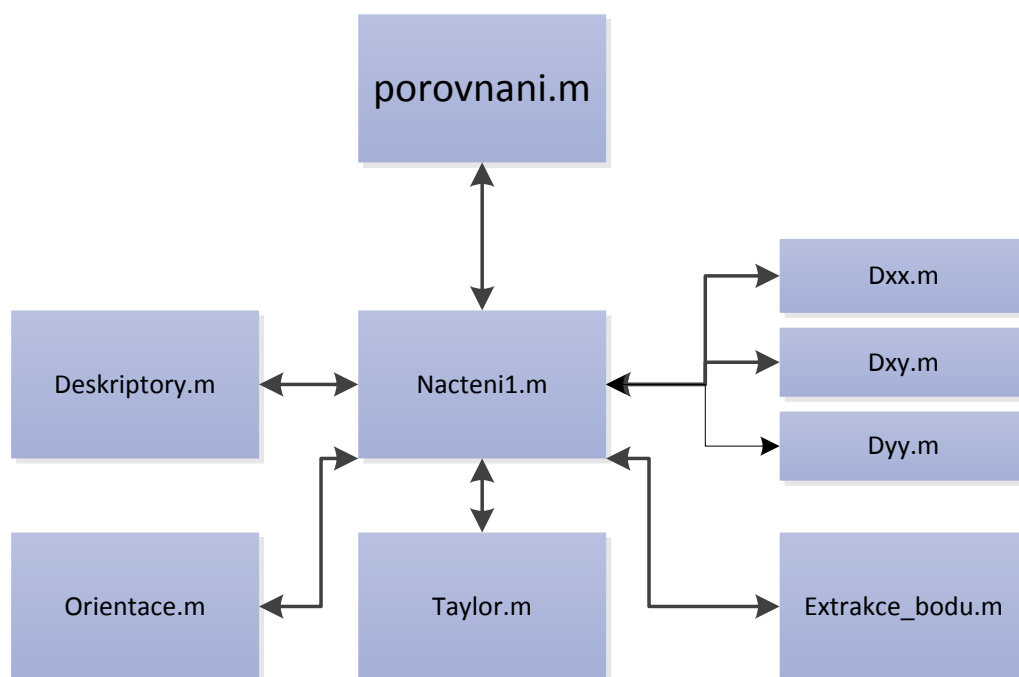
7 Implementace metody SURF

7.1 Programové prostředí

Matlab je vyšší programovací jazyk a interaktivní prostředí pro matematické modelování a vizualizace. Jako každý software má Matlab své pro a proti. Velkou výhodou je zde velké množství před programovaných knihoven plných velmi užitečných funkcí, které uživateli usnadňují práci z hlediska času a optimalizace. Dále zde není nutné definovat proměnné. Proměnná se definuje přiřazením první hodnoty a svůj typ může měnit v průběhu celého programu, což na jednu stranu zjednodušuje použití, na druhou stranu se vzniklé problémy s proměnnými mnohdy složitě odhalují. Právě velké množství předpřipravených knihoven a zkušenosti s programováním v tomto jazyce v průběhu dosavadního bakalářského studia byly hlavní argumenty pro zvolení právě tohoto programovacího prostředí pro implementaci zvoleného algoritmu.

7.2 Popis programu

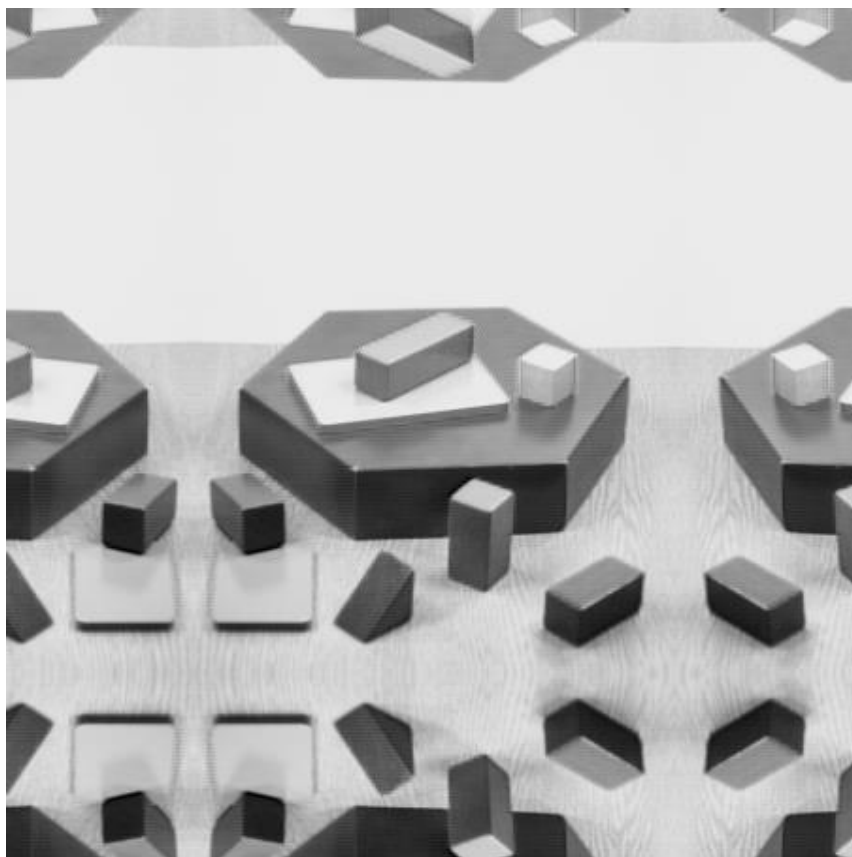
Celé porovnání obrazu se spouští skriptem *porovnani*. Ten načte dva obrazy a pomocí funkce *nacteni1* a jejich podfunkcí vypočítá deskriptory pro oba obrazy. Ty pak mezi sebou porovná na základě euklidovských vzdáleností. Schéma celého programu je na obrázku 9. Jednotlivé funkce budou probrány v další části této práce.



Obrázek 9: Schéma programu

7.2.1 Konvoluce s Gaussovými filtry

Jako první krok pro extrakci bodů je nutné vybudovat scale space pomocí konvolucí s maskami o různých velikostech odpovídajícími diskretizovaným derivacím Gaussových funkcí (viz část 4). Ty využívají předem vygenerovaného integrálního obrazu. Při konvoluci s maskami o různých velikostech se potýkáme s problémem výpočtu konvoluce v krajních bodech, kde maska přesahuje hranice obrazu. Program tak indexuje výpočet mimo matici, čímž vyvolá chybu. To bylo ošetřeno symetrických rozšířením obrazu o rozměr největšího filtru. Tuto operaci ilustruje obrázek 10.



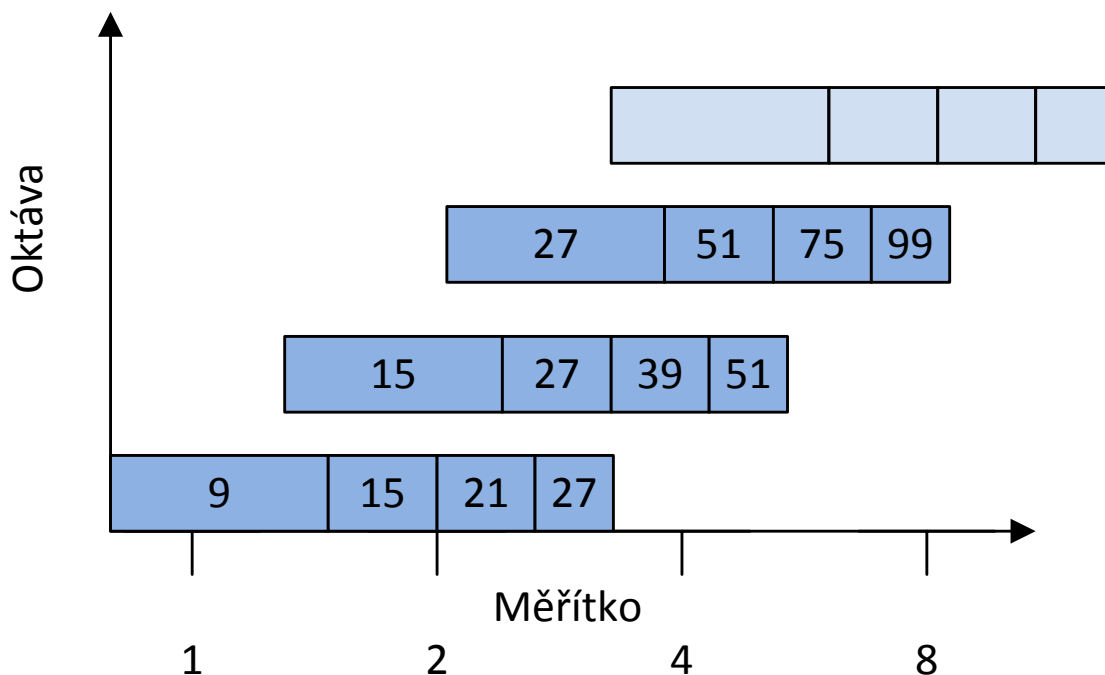
Obrázek 10: Ukázka symetrického rozšíření obrázku jako prevence indexování mimo obraz

Vlastní konvoluci filtrů s integrálním obrazem zajišťují 3 funkce: D_{xx} , D_{yy} a D_{xy} , kde každá funkce zajišťuje konvoluci s maskami tvořenými derivacemi Gaussových funkcí v naznačených směrech. Jako vstupní parametry těchto funkcí vystupují integrální obraz, rozměr konvoluční masky a rozměry původního obrazu. Z těchto parametrů je dále vypočtena konvoluce s daným filtrem. Rozměr konvoluční masky zde má zásadní význam ve smyslu nezávislosti metody na měřítku. Čím větší bude velikost filtru, tím větší bude prostorové zkreslení, které nám simuluje zmenšení obrazu a následné nalezení deskriptoru ve větším měřítku. Podrobněji se tímto budeme zabývat dále. Výstupy těchto funkcí jsou dále normalizovány vzhledem k velikosti filtrů. Tímto postupem vlastně vypočítáme jednotlivé prvky Hessovy matice, ze které se dále počítá determinant. Ten se bude postupně ukládat do 3D matice, kde třetí dimenze bude reprezentovat rozměr filtru, ze kterého byl vypočítán.

7.2.2 Extrakce bodů

Jednotlivé body jsou vyhledávány algoritmem známým pod jménem non-maximum suppression. Jedná se o hledání lokálních maxim determinantu Hessovy matice v $3 \times 3 \times 3$ okolí v jednotlivých oktávách [2] (pro názornost odkazují na obrázek 4). Pod pojmem oktáva si můžeme představit část 3D matice vypočtené v předchozím kroku. Pro ilustraci a představu

konkrétních číselných hodnot je zde obrázek 11. Při implementaci je požito 3 oktáv, což přesně odpovídá přiloženému obrázku.



Obrázek 11: Grafická ukázka generování oktáv. Na ose x je zobrazeno měřítko, osa y reprezentuje oktávu. Čísla v jednotlivých kolonkách reprezentují velikost konvoluční masky. Převzato z [1] a upraveno.

Pokud tento algoritmus aplikujeme přímo na determinant vypočítaný v předchozím kroku, nalezneme všechny významné body definované metodou SURF. Některé budou méně určující, některé více, záleží na nominální hodnotě determinantu v daném bodě. Je na každé konkrétní aplikaci zda vyžaduje velký počet méně stabilních bodů, či menší počet bodů za cenu větší stability. Toto je regulováno prahováním determinantu před hledáním lokálních maxim.

Po nalezení všech bodů si musíme uložit jejich souřadnice jak na x-ové a y-ové ose, což nám dává informaci o pozici bodu v obraze, tak na z-ové ose, kde se dozvídáme informaci o měřítku, ve kterém byl bod nalezen.

Přesnou hodnotu měřítka pak vypočítáme podle vzorce

$$s = \text{velikost použitého filtru} * \frac{\text{měřítko základního filtru}}{\text{velikost základního filtru}} = \text{velikost použitého filtru} * \frac{1,2}{9}. \quad (7.1)$$

Vycházíme z toho, že rozměr základního filtru 9×9 odpovídá měřítku s=1,2 (odpovídá derivaci Gaussovy funkce se $\sigma=1,2$) [5].

7.2.3 Zpřesnění lokalizace bodů 3D Taylorovým polynomem

K této kapitole bylo čerpáno z [4]. Dosud jsme vyhledali významné body a lokalizovali je ve třech souřadnicích, a sice x a y , které uvádí informaci o poloze bodu v původním obraze a souřadnici z , jenž nám podává informaci o měřítku. Všechny tyto informace jsou dány s konečnou přesností, která je pro x -ovou a y -ovou souřadnici dána vzorkováním původního obrazu a pro z -ovou souřadnici taktéž vzorkováním původního obrazu a navíc přesností aproximace použitých box filtrů. Přesnost lokalizace bodů klesá s velikostí měřítka, jelikož při použití větších filtrů vlastně podvzorkováváme (rozmazáváme) původní obraz. Zpřesnění lokalizace bodů na subpixelovou úroveň lze provést pomocí interpolace 3D Taylorovým polynomem. Aproximace se provádí pomocí vzorce

$$L(x) = L + \frac{\partial L^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 L}{\partial x^2} x, \quad (7.2)$$

kde $x = (x, y, s)^T$ je souřadnice bodu x ve scale space a $L(x)$ je samotné scale space tvořeno determinanty Hessovy matice. Všechny parciální derivace jsou zde počítány aproximací za použití rozdílu pixelů. Konečná subpixelová lokalizace je určena jako extrém funkce

$$\hat{x} = -\frac{\partial^2 L^{-1}}{\partial x^2} * \frac{\partial L}{\partial x}. \quad (7.3)$$

V praxi je toto počítáno jako násobení dvou matic, kde $\frac{\partial^2 L}{\partial x^2}$ je 3×3 Hessova matice a $\frac{\partial L}{\partial x}$ je vektor parciálních derivací.

$$\frac{\partial^2 L}{\partial x^2} = \begin{bmatrix} d_{xx} & d_{yx} & d_{sx} \\ d_{xy} & d_{yy} & d_{sy} \\ d_{xs} & d_{ys} & d_{ss} \end{bmatrix} \quad (7.4)$$

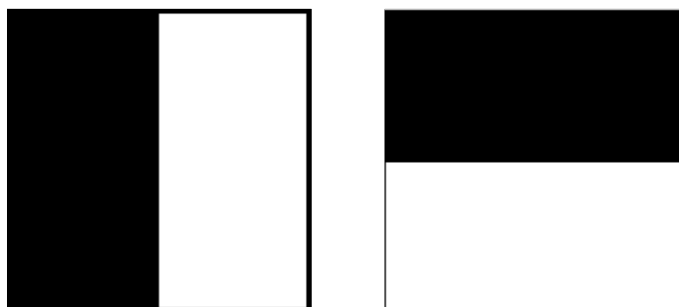
a

$$\frac{\partial L}{\partial x} = \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix}. \quad (7.5)$$

Zde d_x je parciální derivací scale space podle x a d_{xx} je druhou parciální derivací podle x . Po umocnění na méně prvou a vynásobení pak dostaneme vektor tří hodnot, které když přičteme k jednotlivých souřadnicím, dosáhneme zpřesnění lokalizace bodu vzhledem k nalezenému extrému ve scale space. Pro tento krok byla upravena funkce převzata z [6]. Tato funkce vypočítá rozdíl mezi souřadnicí opravdového maxima a nalezeného bodu. Pokud je rozdíl v některé souřadnici větší jak 0,5, znamená to, že bod byl nesprávně lokalizován a je proto odstraněn. Je zde totiž možnost, že pro toto maximum byl vybrán jiný bod, který byl nalezen v jeho větší blízkosti. Po zpřesnění lokalizace obou bodů bychom pak měli dva stejné body.

7.2.4 Přiřazení orientace

Pro nezávislost metody na měřítku je nutno každému bodu přiřadit orientaci, která je vypočítána z jeho okolí. Pro výpočet jsou použity 2D Haarovy vlny jako konvoluční masky. Jelikož jsou počítány pouze ve směru x a y, je zde možno požit integrálního obrazu. Grafické znázornění Haarových vln je na obrázku 12.



Obrázek 12: 2D Haarovy vlny, vlevo ve směru x, vpravo ve směru y. Bílá oblast představuje váhu -1, černá oblast váhu 1.

Tyto konvoluční masky jsou aplikovány v kruhovém okolí bodu o velikosti 6s, kdy s znamená měřítko, ve kterém byl bod nalezen. Orientace je počítána z vygenerovaného scale space, konkrétně z toho měřítka, kde byl bod detekován. Odezvy na tyto filtry jsou pak váhovány Gaussovou funkcí, která má střed ve vyšetřovaném bodě a $\sigma=2,5s$ [3]. Pro rozhodnutí o orientaci se zvažuje kruhová oblast kolem každého bodu, ve které jsou zaznamenávány odezvy na Haarovy vlny do histogramu o 6 binech (1bin odpovídá 60°), přičemž odezvy jsou váhovány Gaussovou funkcí umístěnou ve vyšetřovaném bodě. V histogramu pak největší pík reprezentuje orientaci významného bodu.

Pro tento krok byly opět převzaty funkce z [6]. Funkce `nacteni1.m` zavolá funkci `orientace`, které předá vstupní parametry strukturu body a vypočtené determinanty (scale space). K funkci `orientace` pak patří funkce `IntegrallImage_HaarX` a `IntegrallImage_HaarY`, které v cyklu počítají odezvy na Haarovy vlny. Vypočítané orientace jsou uloženy do struktury body a předány jako výstupní parametr.

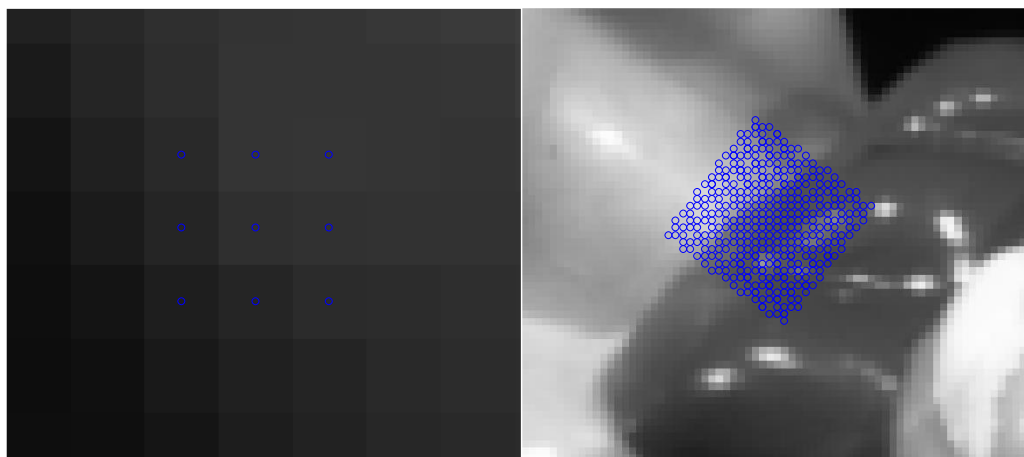
Algoritmus SURF má několik verzí. Jedna z nich je tzv. U-SURF (upright version), která je závislá na rotaci. Tato verze přeskakuje krok určování orientace. Publikace [1] uvádí, že odolá rotaci $\pm 15^\circ$. V další části práce jsou obě verze testovány z hlediska porovnávání medicínských obrazů.

7.2.5 Výpočet deskriptorů

Z teoretického hlediska je výpočet deskriptorů probrán v kapitole 4.1.2. V přiloženém programu toto zajišťuje funkce `deskripty`. Jako vstupní parametry zde vystupuje struktura body, která obsahuje všechny doposud zjištěné informace o nalezených významných bodech a originální obraz, ze kterého jsou deskripty počítány. Při počítání deskriptorů blízko okrajů

obrazu vzniká riziko indexování mimo matici, ve které je obraz uložen. To je ošetřeno rozšířením matice o nuly.

U verze nezávislé na rotaci zde pak vzniká problém natočení jak okolí, ze kterého je deskriptor počítán, tak samotných Haarových vln. Při malých měřítkách jsou jednotlivé konvoluční masky tak malé, že zachycují pouze 9×9 pixelů, což zúží možnosti natočení pouze na násobky $\frac{\pi}{2}$. Tím mohou vznikat nepřesnosti při porovnávání natočených obrazů, zejména při lícování bodů, které byly nalezeny v malých měřítkách.



Obrázek 13: Ukázka natočení masek pro výpočet Haarových vln. V obou případech bylo použito natočení o 116° . Na obrázku vlevo je použita maska vztahující se k měřítku $s=1$, vpravo k měřítku $s=10$. Z důvodu limitace původním vzorkováním obrazu lze natočit konvoluční masku v malých měřítkách pouze o násobky 90° .

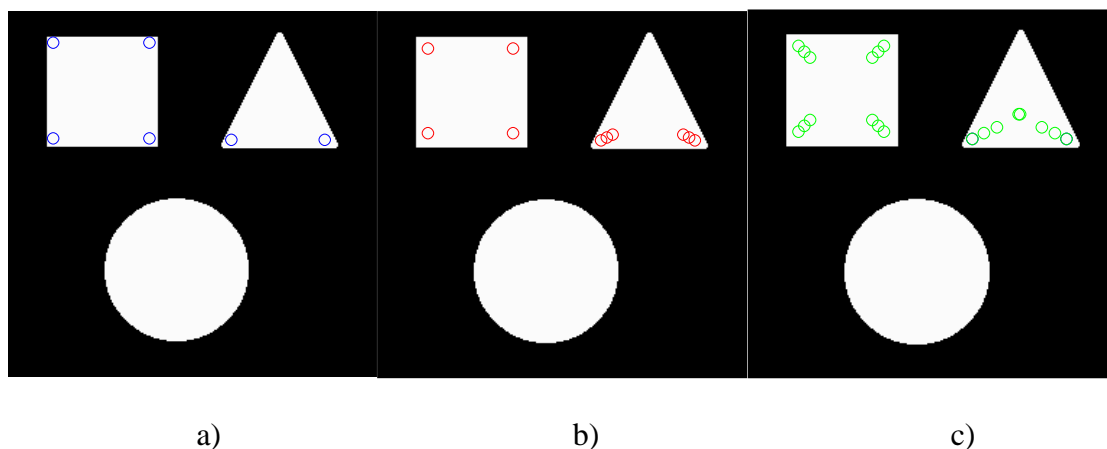
7.2.6 Porovnání deskriptorů

Pro porovnání deskriptorů vzájemně mezi dvěma obrazy bylo použito metody „2nd nearest neighbour“ zveřejněného v publikaci [2]. Pro všechny možné dvojice deskriptorů mezi dvěma obrazy jsou počítány Euklidovské vzdálenosti. Shoda mezi dvěma deskriptory nastane, pokud vzdálenost nejbližšího deskriptoru menší jak 0,7 násobek druhé nejmenší vzdálenosti. Tato metoda je výhodná funkčně i z důvodu její jednoduché myšlenky a následné implementace.

8 Testování na modelových datech

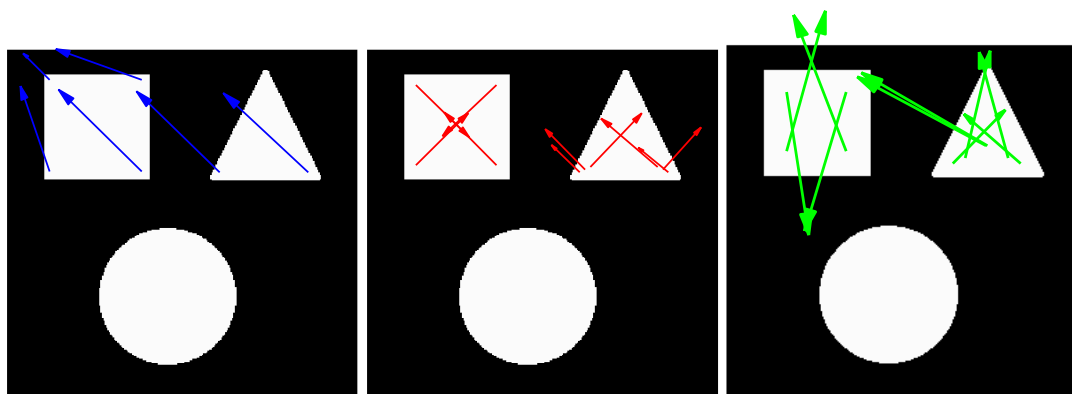
Pro testování základních vlastností použitého algoritmu je nejlepší použít umělé vytvořené obrazy, které obsahují základní prvky, jako jsou úhly o různých velikostech, hrany natočené v různých úhlech od vertikální osy obrazu, různé přechody intenzit atd. Pro toto testování budeme používat obrazy z kapitoly 6.

Pro první testování použijeme obrázek 8 a). Detektor byl nastavován tak, aby v každé oktávě bylo vidět několik reprezentativních bodů.



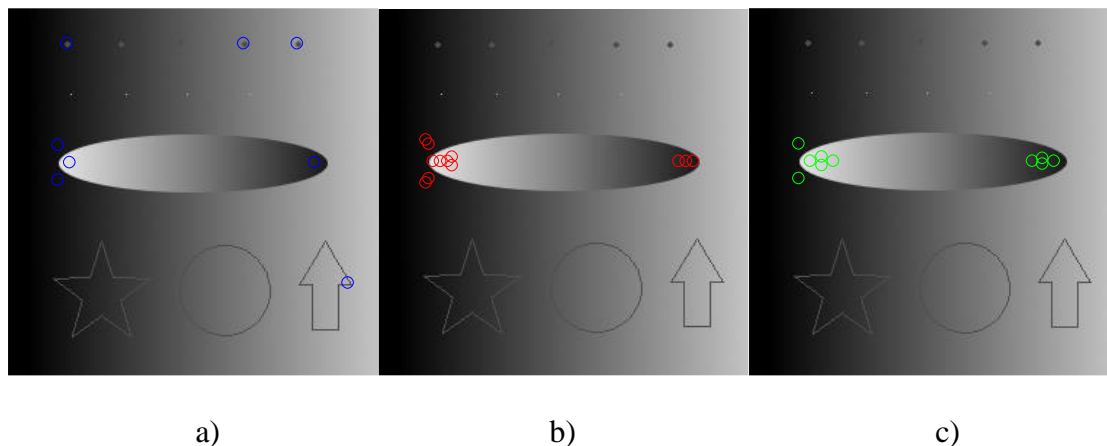
Obrázek 14: Ukázka extrahování bodů v jednotlivých oktávách na základním obrázku se dvěma intenzitami (0 a 255). Na obrázku a) jsou zobrazeny body nalezeny v první oktávě, na obrázku b) jsou body extrahované z druhé oktávy a na obrázku c) pak ze třetí oktávy.

Obrázek 14 ukazuje, že algoritmus nejvíce reaguje na rohy, tedy spojnice dvou hran. Nejedná se však o detektor rohů, který by je přesně lokalizoval, ale o detektor významných bodů, který bere v potaz i určité okolí bodu. Lze zde také pozorovat rozdíly mezi body extrahovanými z různých oktáv. Body nalezeny v první oktávě jsou lokalizovány pouze v těsné blízkosti rohů obrázku, zatímco ve vyšších oktávách jsou body lokalizovány dále od rohů.



Obrázek 15: Ukázka orientace jednotlivých bodů. Počátek šipek reprezentuje vždy významný bod. Natočení šipek odpovídá zjištěné orientaci významných bodů, poměr délek šipek pak odpovídá měřítku, ve kterém byly body nalezeny (poměr odpovídá vždy pouze na jednom obrázku, z důvodu lepšího zobrazení).

Na obrázku 15 můžeme pozorovat orientaci jednotlivých bodů znázorněnou pomocí šipek. Porovnáme-li lokalizaci bodů nalezených v různých oktávách, liší se pouze minimálně. Srovnáme-li ale jejich orientace, liší se značně. Je to dáno tím, že výpočet orientace je počítán z okolí, jehož velikost je úměrná měřítku, ve kterém byl bod nalezen.



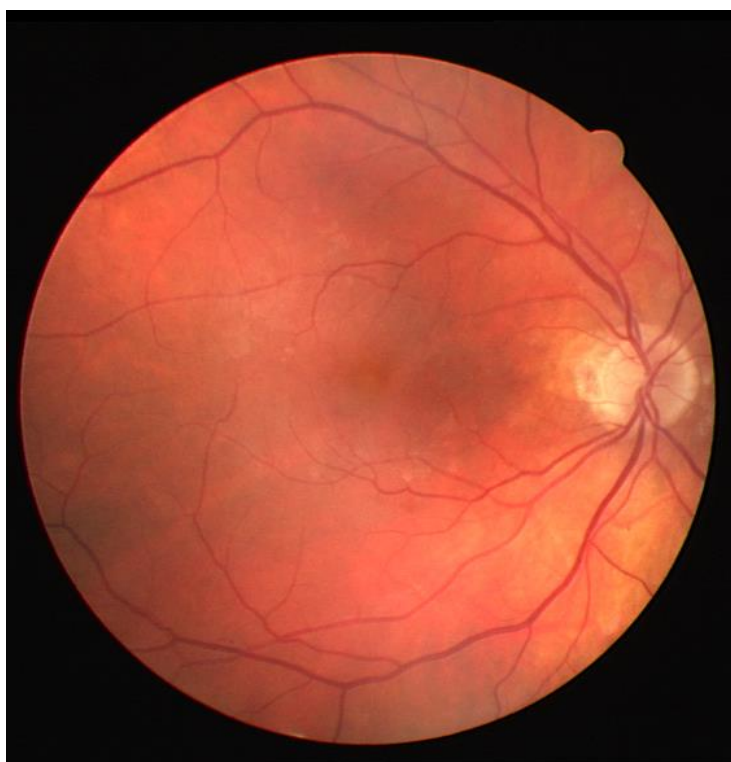
Obrázek 16: Ukázka extrahování bodů v jednotlivých oktávách. Na obrázku a) jsou body extrahované z první oktávy, na obrázku b) z druhé oktávy a na obrázku c) ze třetí oktávy.

Obrázek 16 názorně ukazuje místa, které jsou brány jako významné body. Jsou to hlavně místa s náhlou změnou intenzity. V obrázku 16 a) je vidět, že při hledání bodů v první oktávě se bere v potaz pouze to nejmenší okolí, tím pádem jsme schopni detekovat ty nejmenší detaily. Naopak při detekci bodů ze třetí oktávy jsou malé detaily zanedbány, soustředíme se zde především na změny intenzity ve větším měřítku. Toho je pak využíváno při srovnávání obrazů o různých měřítkách, deskriptory z nižších oktáv jednoho obrazu jsou pak párovány s deskriptory vyšších oktáv obrazu druhého.

Co se týče detekce jednotlivých bodů, je zde poznat, že algoritmus dává přednost vysoce kontrastním hranám před nízko kontrastními rohy. Při snižování prahu determinantu totiž budou nalezeny další body po krajích elipsy, ale nebudou nalezeny žádné body například na hranách hvězdičky.

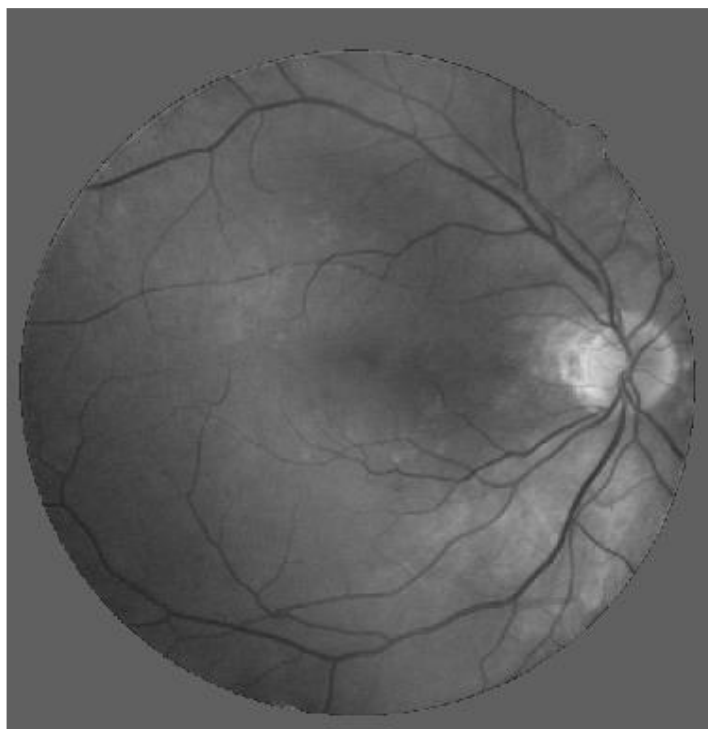
9 Testování na reálných datech

Jak bylo zmíněno výše, použitý algoritmus byl zveřejněn jako odolný proti rotaci, změně měřítka, osvětlení a dalších parametrů. Jelikož se chceme zabývat porovnáváním medicínských obrazů, budeme tyto vlastnosti testovat na databázi fotek sítnice z fundus kamery. Databáze obsahuje 20 snímků v RGB režimu o velikosti 594×791 pixelů a je dostupná z [18]. Jako hlavní parametr se vždy uvádí úspěšnost detekce, což značí kolik procent ze všech spárovaných bodů bylo spárováno správně.



Obrázek 17: Ukázka snímku z fundus kamery. Převzato z [18].

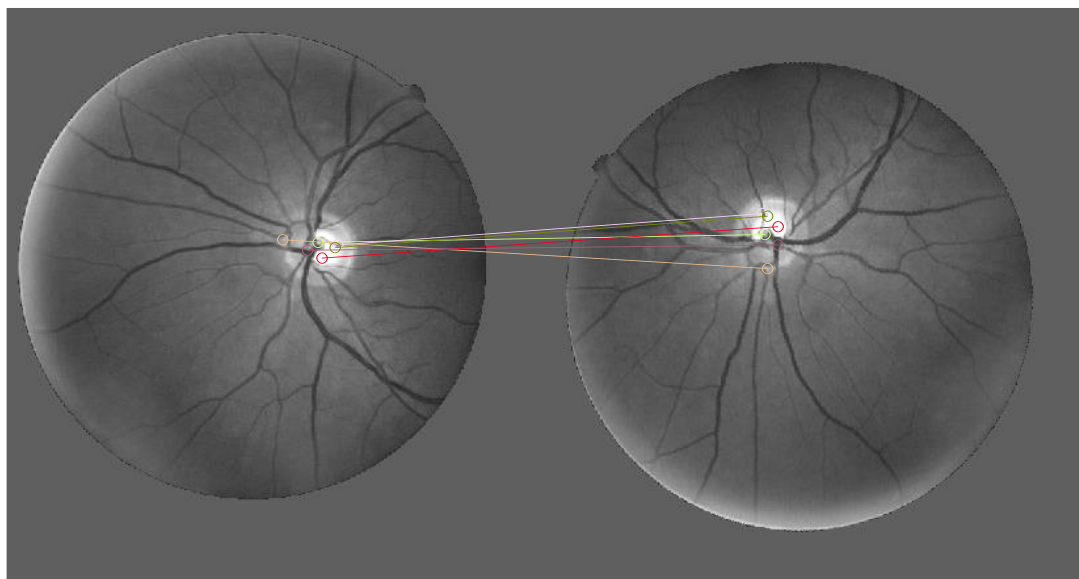
Před porovnáním snímků z fundus kamery diskutovaným programem je nutno provést několik úvodních úprav. Algoritmus SURF pracuje se šedotónovými obrazy, proto je nutné původní snímek převést z RGB režimu, což bylo pro toto testování provedeno výběrem pouze zeleného kanálu. Další překážkou aplikaci algoritmu je černé okolí obrazu. Detektor hledá významné body především na místech s náhlou změnou intenzity, což představuje právě přechod mezi okolím s intenzitou blízkou nule a samotným zobrazením sítnice. Jako významné body jsou potom označeny pixely na tomto rozhraní, které pak mají podobné okolí, tudíž i deskriptory, a jsou si nesprávně podobné. Navíc, na obrazu sítnice je asi největší oblastí zájmu optický disk, proto je nutné detekci směřovat právě k tomuto místu. Následky tohoto problému byly minimalizovány nahrazením okolí o nízkých intenzit hodnotou mediánu vypočítaného z toho samého obrázku ořezaného z každé strany o 100 pixelů. Po těchto úpravách je snímek připraven k detekci.



Obrázek 18: Ukázka snímku sítnice připraveného pro detekci.

9.1 Testování odolnosti vůči rotaci

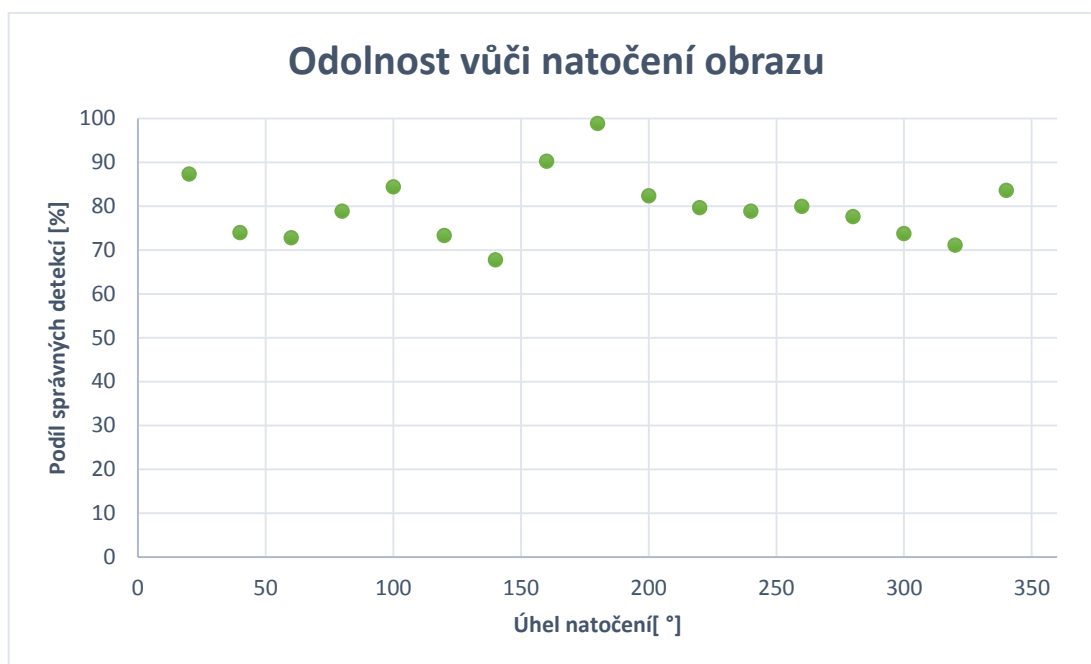
Odolnost vůči rotaci byla testována na celé databázi dvaceti snímků, stejně jako i další parametry uvedené dále v textu. Každý snímek byl nejprve upraven podle postupu popsaného výše a byl postupně natáčen o násobky 20° . Pro každý snímek bylo individuálně nastavováno prahování determinantu tak, aby bylo ve výsledku detekováno 100 až 200 významných bodů. Při tomto testování nebyla použita funkce *Taylor*, tedy interpolace bodů Taylorovým polynomem, jelikož obrazy nebyly nijak deformovány, co se týče jejich rozměrů. Vynechání této funkce se projevilo mírnými nepřesnostmi v lokalizaci jednotlivých bodů (maximálně posun o cca 5 pixelů), které byly při vyhodnocování tolerovány. Ukázka správné detekce na obrazu natočeném o 100° je na obrázku 19. Údaje o výsledcích testování jsou uvedeny v tabulce 1, v grafu 1 je vynesena závislost úspěšnosti detekce na úhlu natočení.



Obrázek 19: Ukázka správné detekce na obrazu natočeném o 100°. Jako shoda bylo

Tabulka 1: Výsledky testování odolnosti detekce vůči rotaci obrázku

Úhel natočení	Celkem správných detekcí	Celkem nesprávných detekcí	Procento úspěšnosti detekce	Průměrný počet správných detekcí na jeden snímek	Průměrný počet nesprávných detekcí na jeden snímek
20	97	14	87,39	4,85	0,70
40	54	19	73,97	2,70	0,95
60	51	19	72,86	2,55	0,95
80	71	19	78,89	3,55	0,95
100	65	12	84,42	3,25	0,60
120	44	16	73,33	2,20	0,80
140	40	19	67,80	2,00	0,95
160	74	8	90,24	3,70	0,40
180	262	3	98,87	13,10	0,15
200	75	16	82,42	3,75	0,80
220	47	12	79,66	2,35	0,60
240	56	15	78,87	2,80	0,75
260	64	16	80,00	3,20	0,80
280	66	19	77,65	3,30	0,95
300	45	16	73,77	2,25	0,80
320	42	17	71,19	2,10	0,85
340	102	20	83,61	5,10	1,00



Graf 1: Závislost úspěšnosti detekce na úhlu natočení

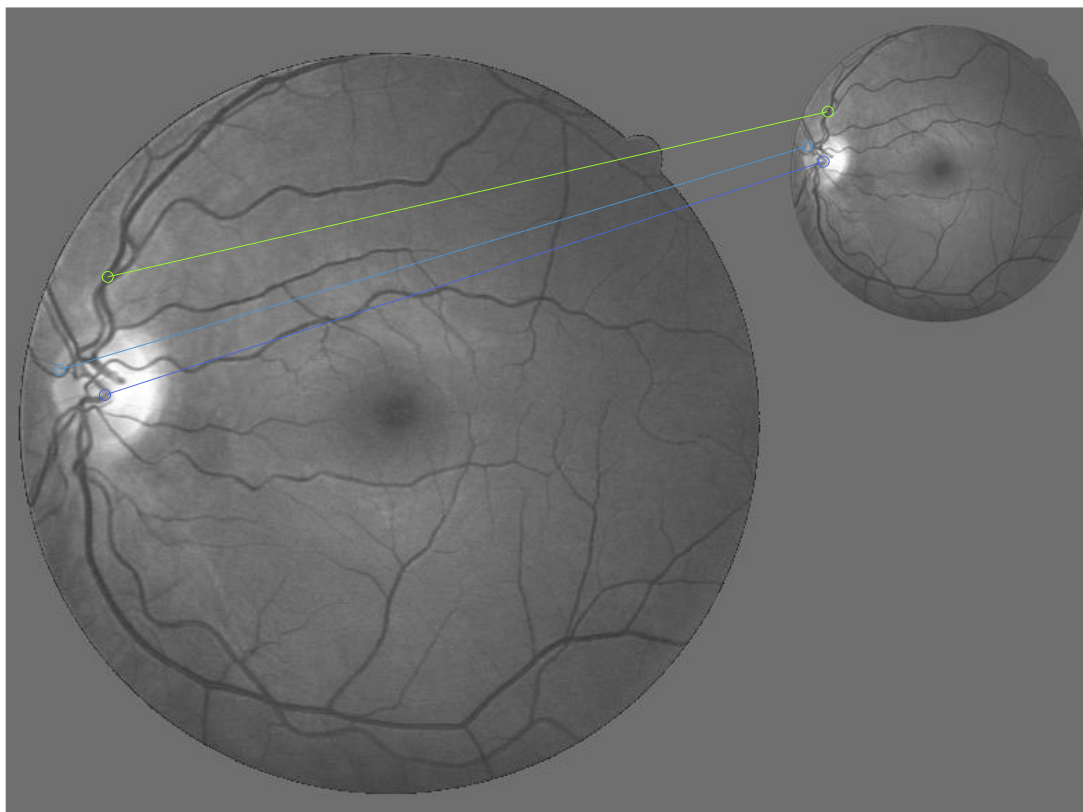
Z grafu 1 je patrné že algoritmus detekuje s dobrými výsledky při natočení o násobky 90° z důvodu popsaného v kapitole 7.2.7, a sice nemožnost natočení malých konvolučních masek o úhly jiné než násobky 90° . Nejlépe však detekuje při natočení obrazu o 180° s účinností 98%. Naopak nejhorších výsledků se dosahovalo při natočení o 140° , i tehdy ale úspěšnost detekce neklesla pod 65%.

Celkovým nedostatkem při detekci natočených obrazů je fakt, že celkový počet detekovaných párů je velmi malý. Celkový průměr správně detekovaných párů na jedno porovnání je 3,69 a nesprávně detekovaných 0,76. Při testování se také v několika případech stalo, že nebyla detekována shoda žádná, nebo bylo detekováno více párů nesprávných než správných. Takových případů ale nastalo méně než 2%.

9.2 Testování účinnosti při změně měřítka

Přiřazování orientace každému bodu před počítáním deskriptorů činí tento algoritmus odolný proti rotaci. Tato vlastnost byla testována opět na databázi [18], kdy byl každému obrazu a upravena velikost s měřítka postupně od 0,4 do 2 s krokem 0,2. Takto upravený obraz byl srovnáván se svým originálem. Při porovnávání byla zapnuta funkce *Taylor*, tedy interpolace Taylorovým polynomem, což výrazně zvýšilo prostorovou lokalizaci jednotlivých bodů, ale také snížilo počet detekovaných bodů. Prahování determinantu bylo nastaveno vždy tak, aby bylo extrahováno vždy 200 bodů s nejvyšší hodnotou determinantu. Celkový počet detekovaných bodů byl pak omezen funkcí *Taylor*, která odstraní nesprávně lokalizované body. Jelikož nebyl obraz po celou dobu nijak natáčen bylo použito metody U-SURF, která vynechává výpočet orientací jednotlivých bodů. Na obrázku 20 je uveden příklad úspěšné

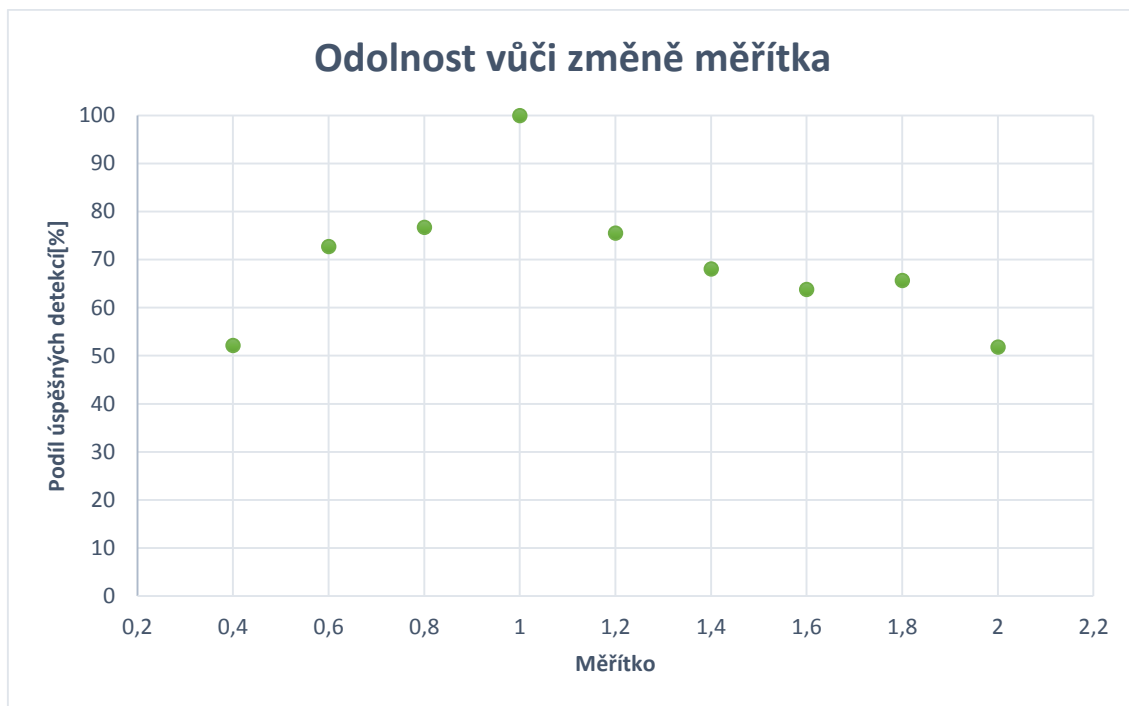
detekce při změně měřítka, výsledky testování jsou uvedeny v tabulce 2, v grafu 2 je pak vynesena závislost poměru úspěšnosti detekce na použitém měřítku.



Obrázek 20: Ukázka správné detekce při změně měřítka na 0,4.

Tabulka 2: Výsledky testování odolnosti vůči rotaci

Měřítko	Počet správných detekcí	Počet nesprávných detekcí	Procento úspěšnosti detekce	Průměrný počet správných detekcí	Průměrný počet nesprávných detekcí
0,4	12	11	52,17	0,60	0,55
0,6	24	9	72,73	1,20	0,45
0,8	66	20	76,74	3,30	1,00
1,2	102	33	75,56	5,10	1,65
1,4	49	23	68,06	2,45	1,15
1,6	30	17	63,83	1,50	0,85
1,8	23	12	65,71	1,15	0,60
2	14	13	51,85	0,70	0,65



Graf 2: Graf závislosti poměru úspěšných detekcí na použitém měřítku

Z grafu 2 lze vyčíst, že úspěšnost správné klesá s hodnotou měřítka vzdalující se od jedničky. Hodnota 100% úspěšných detekcí při měřítku 1 byla přidána bez testování a vyplývá z jednoduché úvahy. Při hledání korespondencí mezi dvěma identickými obrazy budou nalezeny na každém obrazu identické deskriptory, mezi nimiž bude Euklidovská vzdálenost 0, tím pádem budou spárovány.

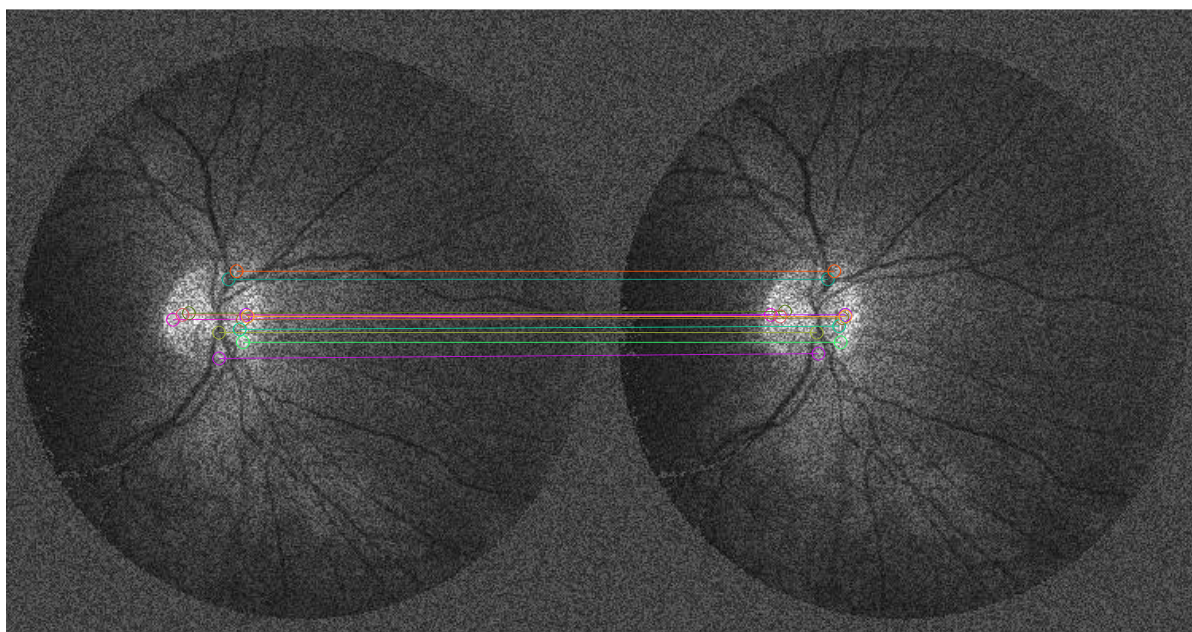
Ověřili jsme tedy, že implementovaná metoda je odolná vůči rotaci a i při dvojnásobném zvětšení či zmenšení obrazu dosahuje úspěšnost detekce více jak 50%. Naopak největším problémem je zde velmi malý počet nalezených shodných bodů. Při extrémních měřítkách 0,4 či 2 je to dokonce méně než jeden bod na jedno porovnání. Z toho vyplývá, že při některých výpočtech nebyl nalezen žádný bod, který by byl spárován. Tato situace nastala přesně ve 23,75% případů. Účinnost metody při změně měřítka je dána hlavně počtem generovaných oktáv, ze kterých se extrahují body. Je tedy nutno poznamenat, že tyto hodnoty jsou platné při generování tří oktáv.

9.3 Testování algoritmu při přidání multiplikativního šumu

Algoritmus byl také testován při přidání multiplikativního šumu. Šum byl přidáván funkcí *imnoise* obsaženou v Matlabu. Šum byl přidáván o oběma obrazům a to tak, že ke každém obrazu byl vygenerován šum nový o stejných parametrech. Šum byl počítán podle vzorce [15]

$$J = I + n * I, \quad (9.1)$$

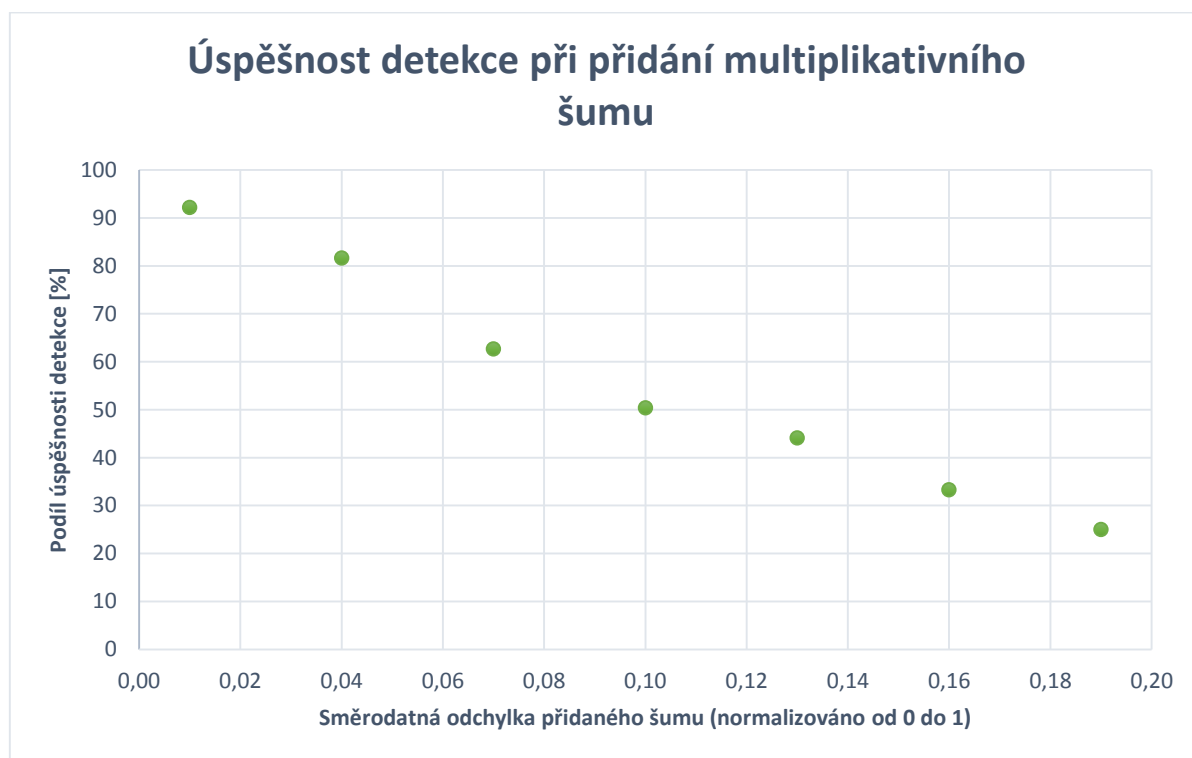
Kdy J je zašuměný obraz a I je původní obraz, n je pak šum se střední hodnotou rovnou nule a směrodatnou odchylkou normalizovanou od 0 do 1 jako měněný parametr. Při tomto testování byla vypnuta funkce *Taylor* a bylo použito metody U-SURF. Testování bylo vyhodnocováno automaticky a to s tolerancí 5 pixelů v každém směru. Ukázka detekce při zašumění je na obrázku 21, výsledky testování jsou uvedeny v tabulce 3, v grafu 3 je pak vynesena závislost úspěšnosti detekce na směrodatné odchylce přidaného šumu.



Obrázek 21: Ukázka detekce při zašumění multiplikativním šumem s relativní směrodatnou odchylkou 0,07. Podle výše zmíněné tolerance je zde správně detekováno deset

Tabulka 3: Výsledky testování úspěšnosti detekce při přidání multiplikativního šumu

Směrodatná odchylka šumu	Počet správných detekcí	Počet nesprávných detekcí	Procento úspěšnosti detekce	Průměrný počet správných detekcí	Průměrný počet nesprávných detekcí
0,01	354	30	92,19	17,70	1,50
0,04	231	52	81,63	11,55	2,60
0,07	116	69	62,70	5,80	3,45
0,10	68	67	50,37	3,40	3,35
0,13	45	57	44,12	2,25	2,85
0,16	35	70	33,33	1,75	3,50
0,19	21	63	25,00	1,05	3,15

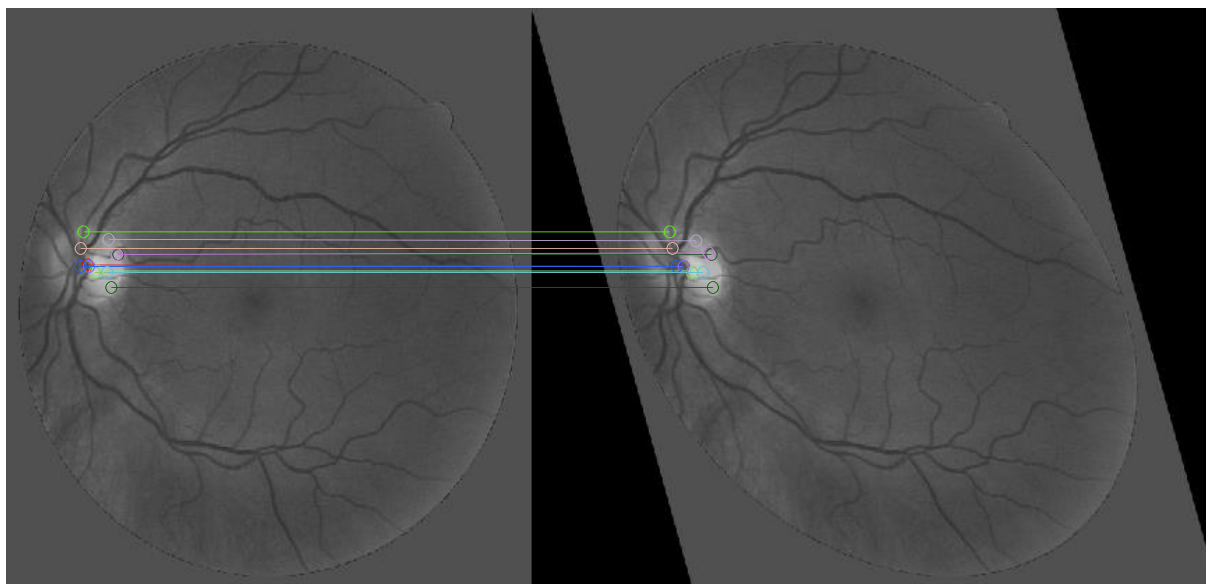


Graf 3: Závislost úspěšnosti detekce na směrodatné odchylce šumu

Z grafu 3 je patrné, že s rostoucí směrodatnou odchylkou šumu klesá úspěšnost detekce. Při přidání šumu se směrodatnou odchylkou 0,01 je úspěšnost detekce 92,19%, zatímco při změně směrodatné odchylky o jeden řád, tedy na 0,1 klesla úspěšnost detekce na pouhých 50,37%. S rostoucí směrodatnou odchylkou klesá taky počet detekovaných bodů, což je při praktickém použití významný parametr.

9.4 Testování úspěšnosti detekce při deformacích obrazu

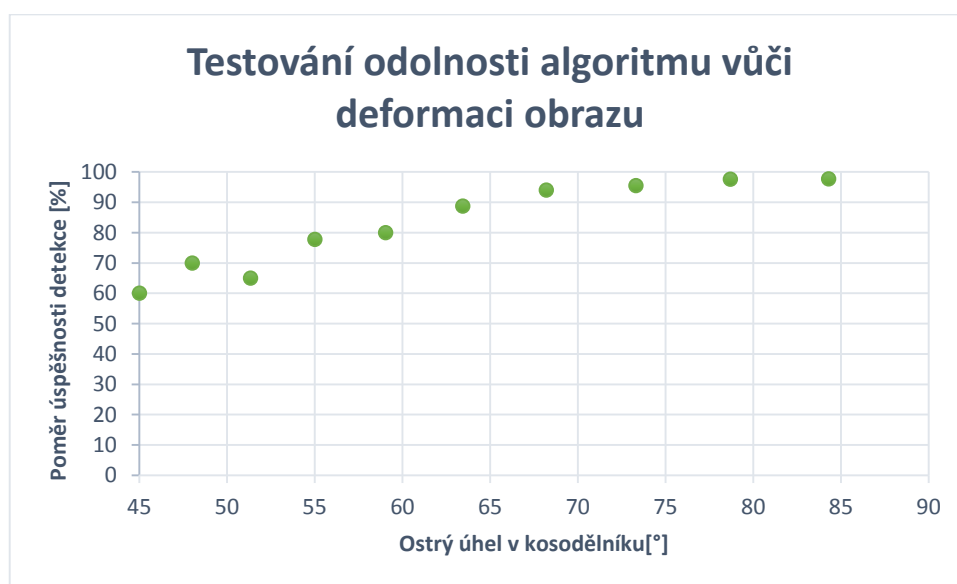
Algoritmus SURF byl původně navržen tak, aby zvládal porovnat i dvě fotky téhož předmětu pořízené s různého úhlu. Tuto vlastnost budeme simulovat deformací obrazu ve smyslu transformace obdélníkového tvaru obrazu na kosodélník. Hlavním kritériem tedy bude úspěšnost detekce v závislosti na vnitřním úhlu kosodélníku. Vždy byly srovnávány body extrahované z původního obrazu s body nalezenými v obrazu deformovaném. Testování probíhalo s verzí U-SURF, při vypnuté funkci *Taylor* a s nastavením prahování determinantu tak, aby bylo na každém obraze nalezeno 100 významných bodů s nejvyšší hodnotou determinantu. Vyhodnocování pak bylo prováděno s lokalizační tolerancí 3 pixelů. Ukázka takovéto detekce je na obrázku 22, výsledné hodnoty vzešlé z tohoto testování jsou uvedeny v tabulce 4, v grafu 4 je pak vynesena závislost úspěšnosti detekce na vnitřním úhlu kosodélníku.



Obrázek 22: Ukázka správné detekce při deformaci obrazu. Vnitřní úhel v kosodélníku je zde $73,3^\circ$, správně bylo spárováno 12 bodů, nesprávně žádný.

Tabulka 4: Výsledné hodnoty vzešlé z testování úspěšnosti detekce při deformaci obrazu

Vnitřní úhel v kosodélníku	Počet správných detekcí	Počet nesprávných detekcí	Procento úspěšnosti detekce	Průměrný počet správných detekcí	Průměrný počet nesprávných detekcí
84,29	378	9	97,67	18,90	0,45
78,69	202	5	97,58	10,10	0,25
73,30	148	7	95,48	7,40	0,35
68,20	94	6	94,00	4,70	0,30
63,43	63	8	88,73	3,15	0,40
59,04	36	9	80,00	1,80	0,45
55,01	28	8	77,78	1,40	0,40
51,34	13	7	65,00	0,65	0,35
48,01	14	6	70,00	0,70	0,30
45,00	9	6	60,00	0,45	0,30



Graf 4: Graf závislosti úspěšnosti detekce na ostrém vnitřním úhlu v kosodélníku deformovaného obrazu.

Z grafu 4 lze vyčíst, že algoritmus je opravdu odolný vůči deformaci obrazu. Úspěšnost při deformaci, kdy vnitřní úhel v kosodélníku byl pouhých 45° dosahovala 60%, což se dá považovat za velký úspěch. Opět zde ale narážíme na problém malého počtu spárovaných bodů, což lze vyčíst z tabulky 4. Už při deformaci při 60° bylo v průměru spárováno méně než dva body, což je pro praktické použití nedostatečné. Na druhou stranu je

nutno vyzvednout metodu „2nd nearest neighbour“, která je používána pro hledání korespondencí mezi deskriptory, že nedovolí velké množství falešně pozitivních detekcí.

10 Závěr

Tato práce byla věnována oboru počítačového vidění, konkrétně vyhledávání významných bodů jako prostředek pro porovnávání obrazů. Byla provedena rešerše, na základě které byly vybrány a shrnuty 4 algoritmy, které byly porovnány z hlediska výpočtu a využití. Do práce byly také zařazeny testovací obrazy, které se využívají při prvotním testování nově implementovaného algoritmu. Z porovnávaných metod byla vybrána metoda SURF, která byla implementována a testována z hlediska porovnání snímků sítnice z fundus kamery.

Algoritmus SURF byl implementován v programovém prostředí Matlab. Při implementaci se používalo především postupu popsáném v publikaci [2], kde je zveřejněno několik verzí tohoto algoritmu. Implementovány byly verze dvě, a sice SURF, která je nezávislá na rotaci, a U-SURF, která není odolná vůči rotaci obrázku. Obě metody používají pro extrakci významných bodů 3 oktávy a jejichmi výstupy jsou deskriptory o 64 hodnotách. Autoři algoritmu uvádějí jeho výhodu zejména v rychlosti výpočtu, která byla dosažena zejména díky použití integrálního obrazu při výpočtu jednotlivých konvolucí. V publikaci [2] uvádějí, že nalezení zhruba 1500 významných bodů na obrázku o rozměrech 600×800 pixelů bylo potřeba pouhých 120 milisekund. Takové rychlosti se nám ale při implementaci nedokázalo zdaleka dosáhnout. Nalezení 1500 významných bodů námi vytvořeným programem na jednom snímku sítnice o velikosti 584×565 pixelů z databáze [22] trvalo 170 sekund, což je nesrovnatelně více. Je to dáno především neefektivním programováním, použitím programovacího prostředí Matlab, a jeho vestavěných funkcí. Nicméně úkolem nebylo navrhnout program co je nejúčinnější z hlediska rychlosti, ale z hlediska použití na medicínských obrazech. V této věci byl algoritmus testován na snímcích z fundus kamery o velikosti 584×565 pixelů. Pro porovnání takovýchto obrazů je neefektivní vyhledávat na každém snímku 1500 bodů, a to jak z hlediska rychlosti, tak z hlediska falešně pozitivních detekcí. Proto bylo při testování použito nastavení takové, aby bylo v každém obraze nalezeno 100 až 200 významných bodů, a to při každém testovaném parametru tak, aby výsledná detekce byla co nejspolehlivější. Při takovémto omezení počtu významných bodů se razantně zvýší rychlost výpočtu. Nalezení cca 150 bodů pak trvá zhruba 25 sekund.

Algoritmus SURF je popisován, jako odolný proti rotaci, změně měřítka, šumu a změně úhlu pohledu na objekt (z angl. 3D viewpoint change). Účinnost algoritmu za těchto podmínek byla proto testována v kapitole 9. Při testování se potvrdila účinnost metody při změně těchto parametrů. Vycházejí z dosažených výsledků se tedy dá prohlásit, že metoda je

vhodná pro porovnávání snímků sítnice z fundus kamery avšak za podmínek předchozího upravení pořízeného obrazu. Tímto se dá prohlásit, že zadání bakalářské práce bylo splněno ve všech bodech.

Seznam literatury

- [1] BAY, Herbert, Andreas ESS, TUYTELAARS a Luc Van GOOL. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*. 2008, č. 110, s. 346-359. Dostupné z: ftp://ftp.vision.ee.ethz.ch/publications/articles/eth_biwi_00517.pdf
- [2] BAY, H., T. TUYTELAARS a L.V. GOOL. Surf: Speeded Up Robust Features. *Ninth European Conference on Computer Vision*. 2006
- [3] BEIS, J.S. a D.G. LOWE. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *Computer Vision and Pattern Recognition*. 1997, s. 1000-1006.
- [4] BROWN, Matthew; LOWE, David G. Invariant features from interest point groups. In: *British Machine Vision Conference, Cardiff, Wales*. 2002. p. 656-665.
- [5] EVANS, Christopher. Notes on the OpenSURF Library. [online]. 2009 [cit. 2013-05-21]. Dostupné z: <http://opensurf1.googlecode.com/files/opensurf.pdf>
- [6] EVANS, Chris. Chris Evans development. [online]. [cit. 2013-05-29]. Dostupné z: <http://www.chrisevansdev.com/computer-vision-opensurf.html>
- [7] GRIMSON, W.E.L. a D.P. HUTTENLOCHER. On the sensitivity of the Hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1990, roč. 12, č. 3, s. 255-274. ISSN 01628828. DOI: 10.1109/34.49052.
- [8] HARRIS, C. a M. STEPHENS. A combined corner and edge detector. In *Fourth Alvey Vision Conference*. 1988, s. 147-151.
- [9] CHEUNG, W. a G HAMARNEH. N-SIFT: N-DIMENSIONAL SCALE INVARIANT FEATURE TRANSFORM FOR MATCHING MEDICAL IMAGES. *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro..* 2007, s. 720-723.
- [10] KUMARI, M.S. a B.H. SHEKAR. On the use of Moravec operator for text detection in document images and video frames. *International Conference on Recent Trends in Information Technology (ICRTIT)*. 2011, s. 910-914.
- [11] LINDBERG, T. Detection with automatic scale selection. *International Journal of Computer Vision*. 1998, 30(2), s. 79-116.
- [12] LIYA, Kong, Huang QINGHUA, Lu MINHUA a Zheng SHUOHE. Image Registration Using SIFT for Extended-Field-of-View Sonography. *4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE)*. 2010, s. 1-4.
- [13] LOWE, David G. UNIVERSITY OF BRITISH COLUMBIA. *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image* [patent]. United States. US Patent, 6,711,293. Uděleno 23.3.2004. Dostupné z: <http://www.cs.ubc.ca/~lowe/keypoints/>
- [14] LOWE, D. Distinctive image features from scale-invariant keypoints, cascade filtering approach. *IJCV*. 2004, č. 60, s. 91-110.
- [15] Oficiální webové stránky MathWorks [online]. [cit. 2013-05-20]. Dostupné z: <http://www.mathworks.com/>

- [16] PARKS, Donovan a Jean-Philippe GRAVEL. Corner Detectors. *Corner Detectors* [online]. [cit. 2012-12-13]. Dostupné z: <http://kiwi.cs.dal.ca/~dparks/CornerDetection/index.htm>
- [17] PARKS, D. a J.P GRAVEL. Corner detection. [online]. [cit. 2012-12-12]. Dostupné z: http://ftp.utcluj.ro/pub/users/nedeveschi/AV/3_CornerDetectors/CornerDetection.pdf.
- [18] STAAL, J.J., M.D. ABRAMOFF, M. NIEMEIJER, M.A. VIERGEVER a B. VAN GINNEKEN. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging*. 2004, č. 4, s. 501-509.
- [19] SUYKENS, J.A.K. a J. VANDEWALLE. Recurrent least squares support vector machines. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*. 2000, roč. 47, č. 7, s. 1109-1114. ISSN 10577122. DOI: 10.1109/81.855471.
- [20] VIOLA, P. a M. JONES. Rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2001, 1-511-1-518.
- [21] XIAO-CHUN, Zou, He MING-YI a Zhao XIN-BO. An efficient medical image tracking algorithm based on motion estimation. *ICME International Conference*. 2009, s. 1-4, 9-11. D
- [22] ZHOU, Ji a Jiaoying SHI. A robust algorithm for feature point matching. *Computers*. 2002, roč. 26, č. 3, s. 429-436. ISSN 00978493. DOI: 10.1016/S0097-8493(02)00086-9.

Seznam použitých zkratek

2D – Two dimensional

3D – Three dimensional

4D – Four dimensional

SIFT – Scale Invariant Feature Transform

SURF – Speeded Up Robust Features

DoG – Difference of Gaussian

CT – Computed Tomography

SVM – Support Vector Machine

U-SURF – Speeded Up Robust Features Upright version